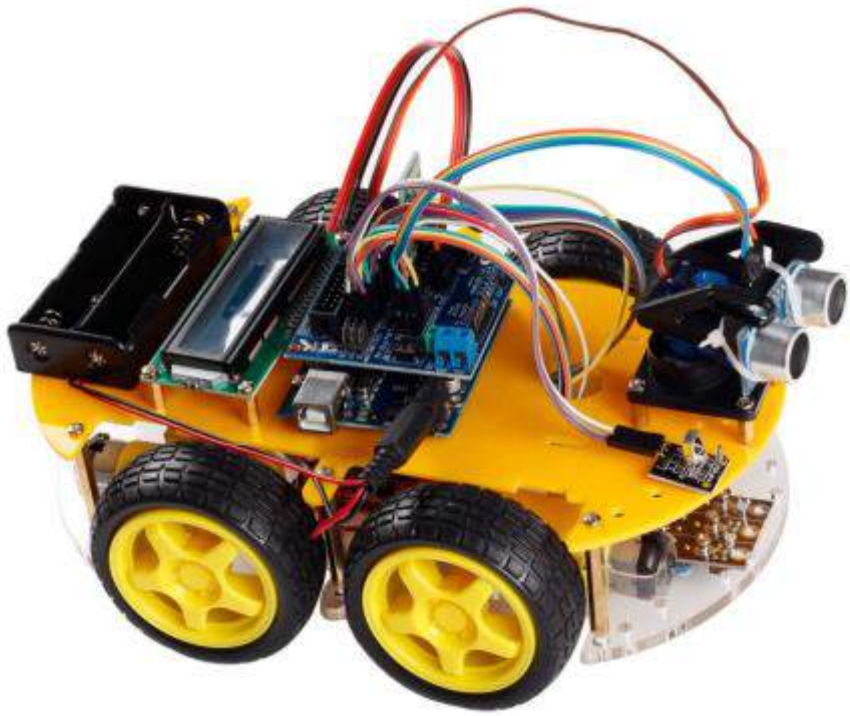
4WD Robot Kit Instruction Manual



1. INTRODUCTION

Bluetooth multifunctional car is a MCU learning and application development system with Arduino single chip atmega-328 as the core. It has the functions of line tracking, obstacle avoidance, infrared remote control. This can be expanded by adding external sensors and modules for more functionality.

2. PARAMETERS

1, motor parameters: voltage range: 6-9V, reduction ratio of 48: 1 ratio
2, motor control uses L298N drive module, real isolation with MCU.

3, three groups of line tracking module, detection of black and white line, higher precision, but also can be used with controlling anti drop.
4, infrared remote control module, make intelligent vehicle control system.
5, ultrasonic module for car obstacle avoidance system.
6, I2C1602 Liquid crystal display makes primary man-machine interface.
7, can access the external voltage of 7~12V, can carry a variety of sensor modules, according to your imagination to achieve various functions.

3. EXPERIMENTAL CURRICULAR INTRODUCTION

1. I2C-LCD1602 LCD test
2. Tracking module adjusting test
3. Ultrasonic module test
4. Servo control routine
5. Infrared remote control test-LCD display
6. L298N motor driver board test
7. Tracking intelligent car
8. Ultrasonic collosion avoidance intelligent car
9. Infrared remote control of intellegent car
10. Four in one (search, obstacle avoidance, infrared remote, Bluetooth remote control) kit.

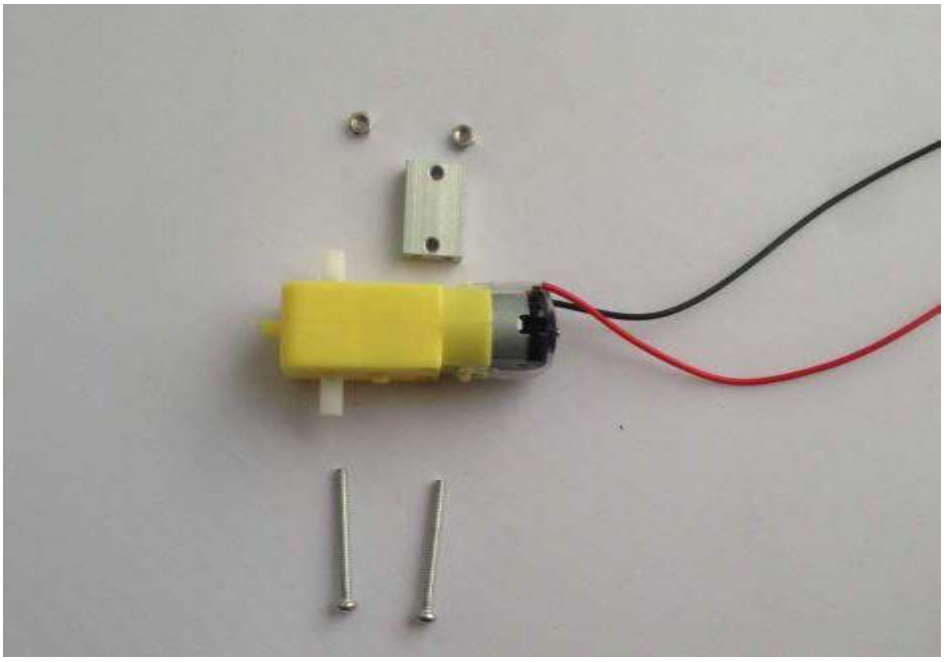
4. LIST

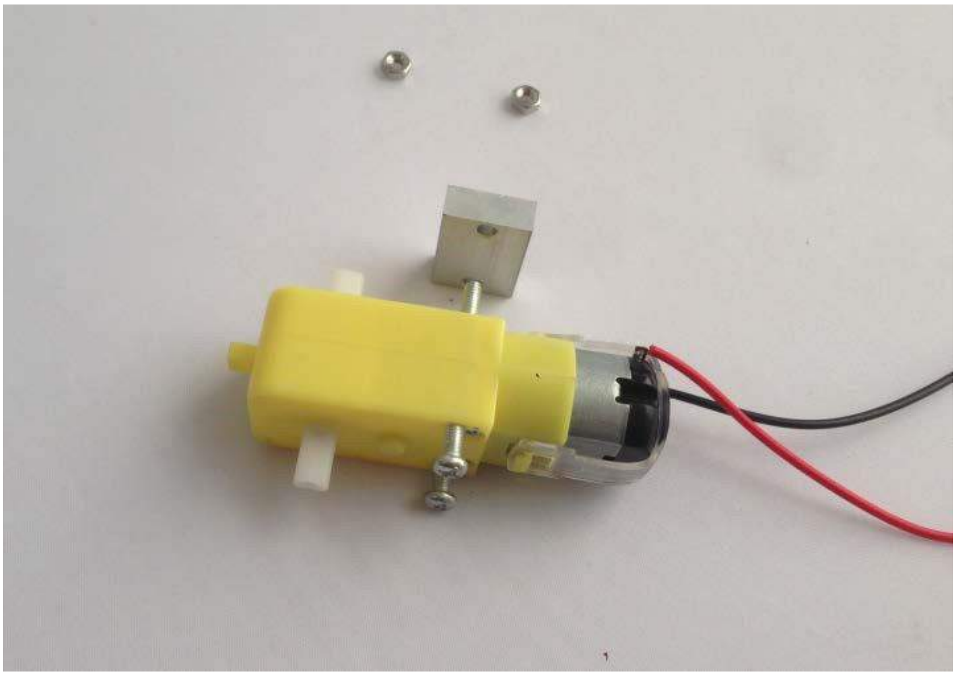
1. 4xGear motor
2. 4xHigh quality tyre
3. 4xThe motor retainer
4. 1x100 * 213 * 5 mm organic glass plate
5. 1x100 * 213 * 5 mm organic glass plate
6. 1x L298N motor driver board
7. 1xThe ARDUINO UNO328 panel
8. The ARDUINO sensor extension boards V5
9. 1x holder
10. 1xservo
11. 1xThe ultrasonic module
12. The I2C - LCD1602 LCD
13. Three groups of tracking module
14. The infrared receiving sensor
15. MCU remote control
16. 1x18650 battery box
17. 18650 2 cell batteries
18. 1x18650 charger
19. 30xThe dupont line
20. 1 meters long USB line 1
21. The copper column 6x35 mm, 3x 20 mm, 6x6 mm

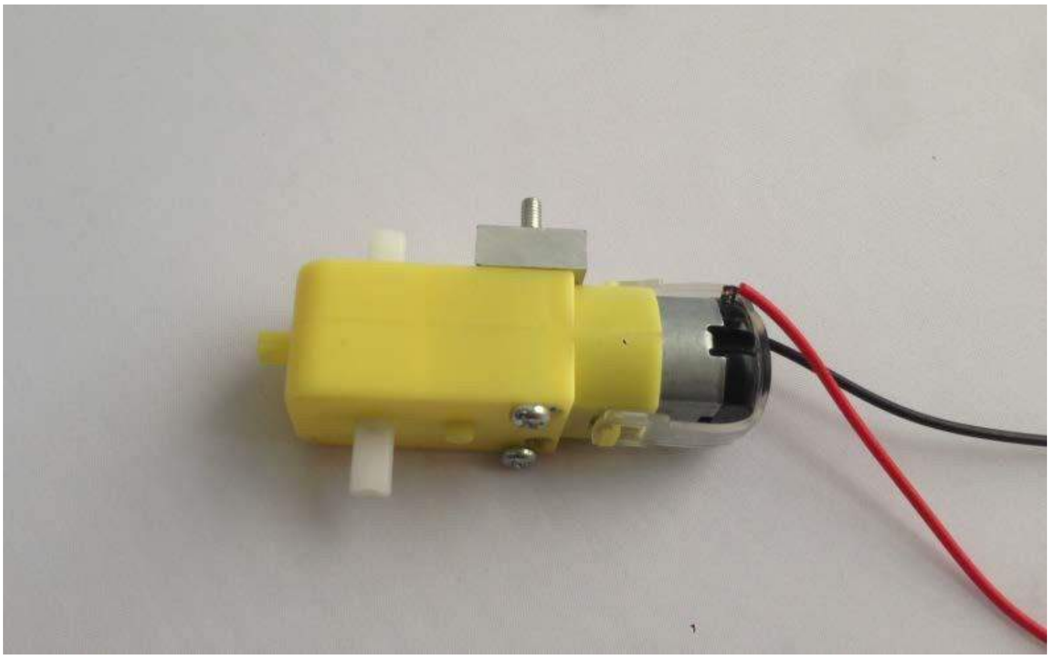
22. 3 MM screw and nut several pieces

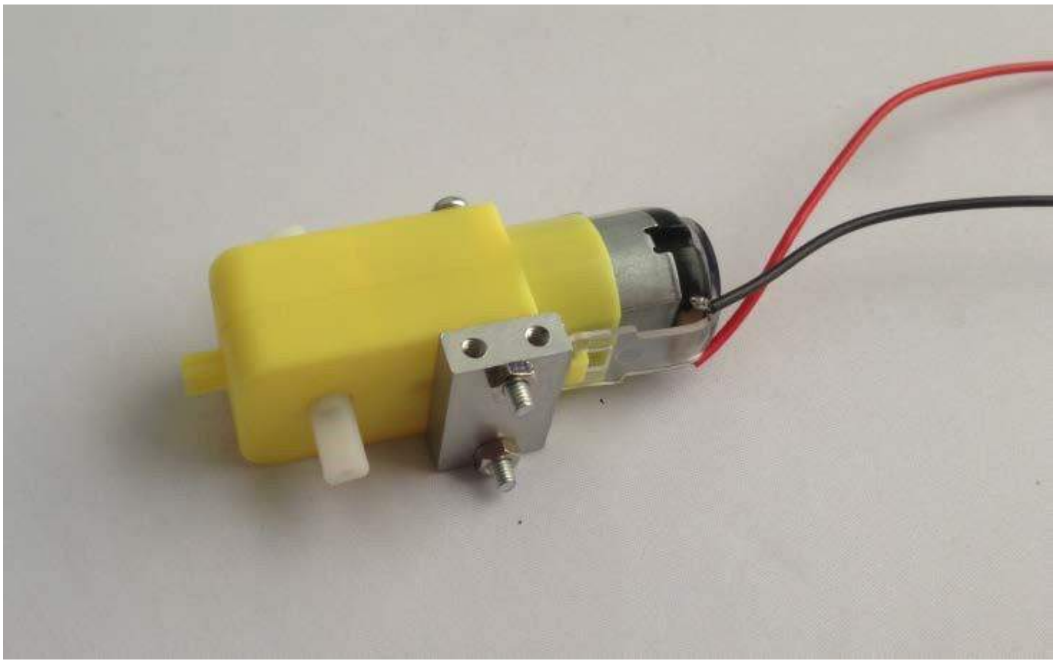
5. HARDWARE INSTALLATION

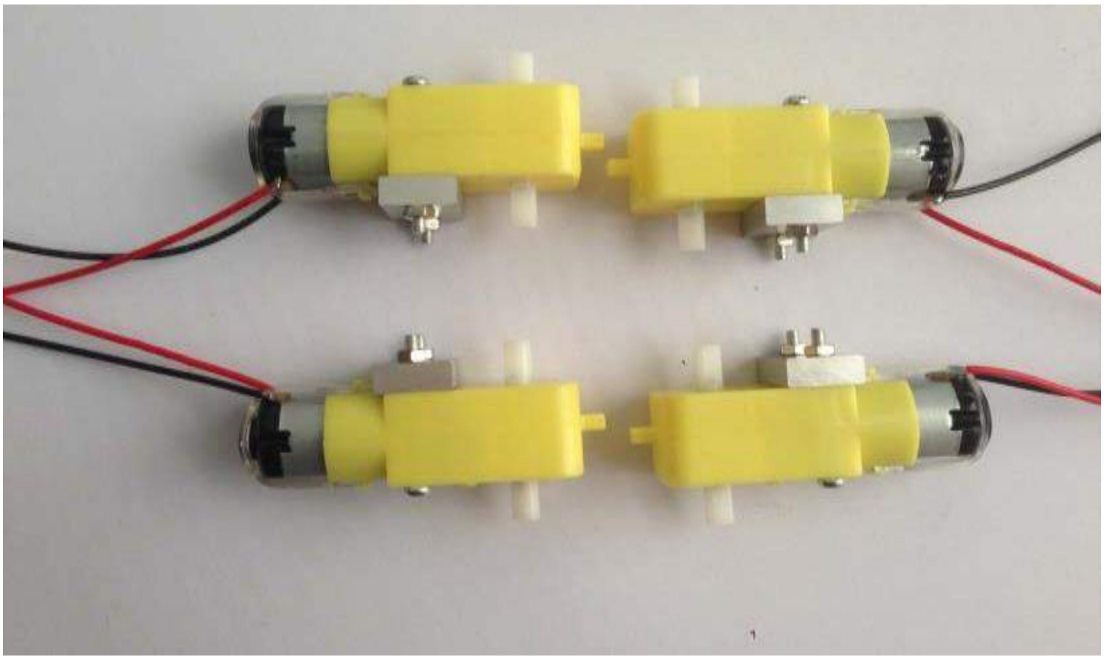
1. MOTOR INSTALLATON



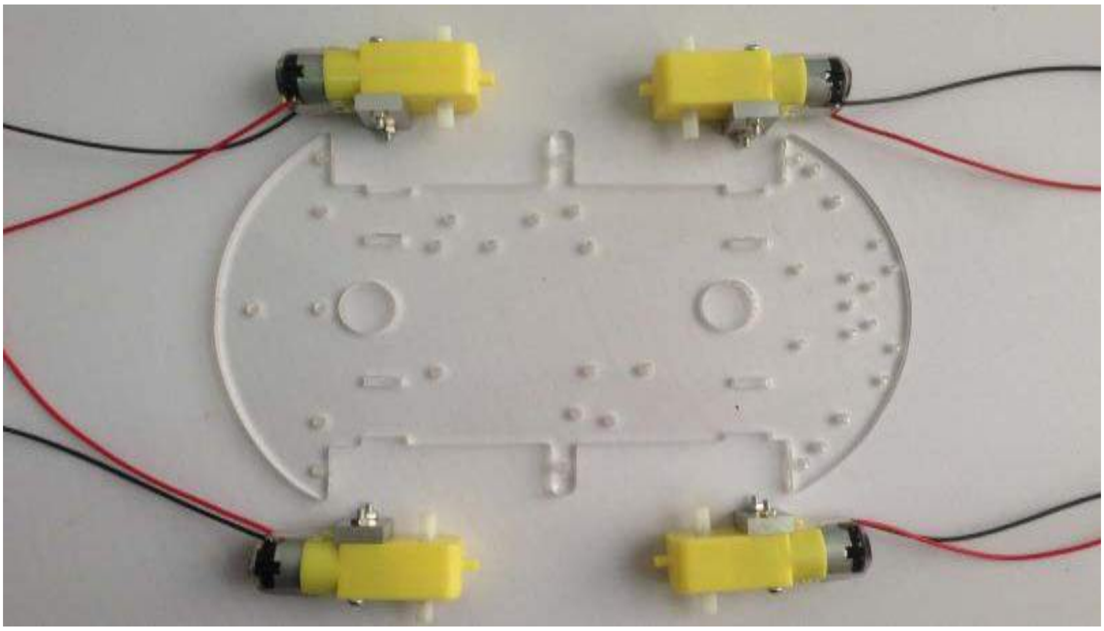


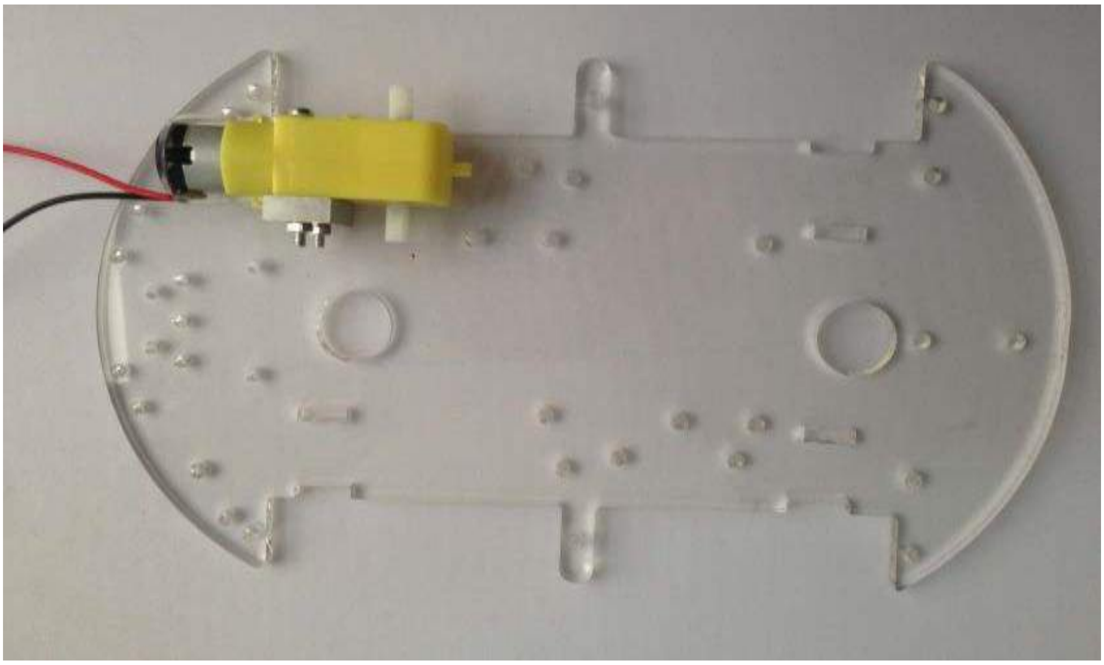


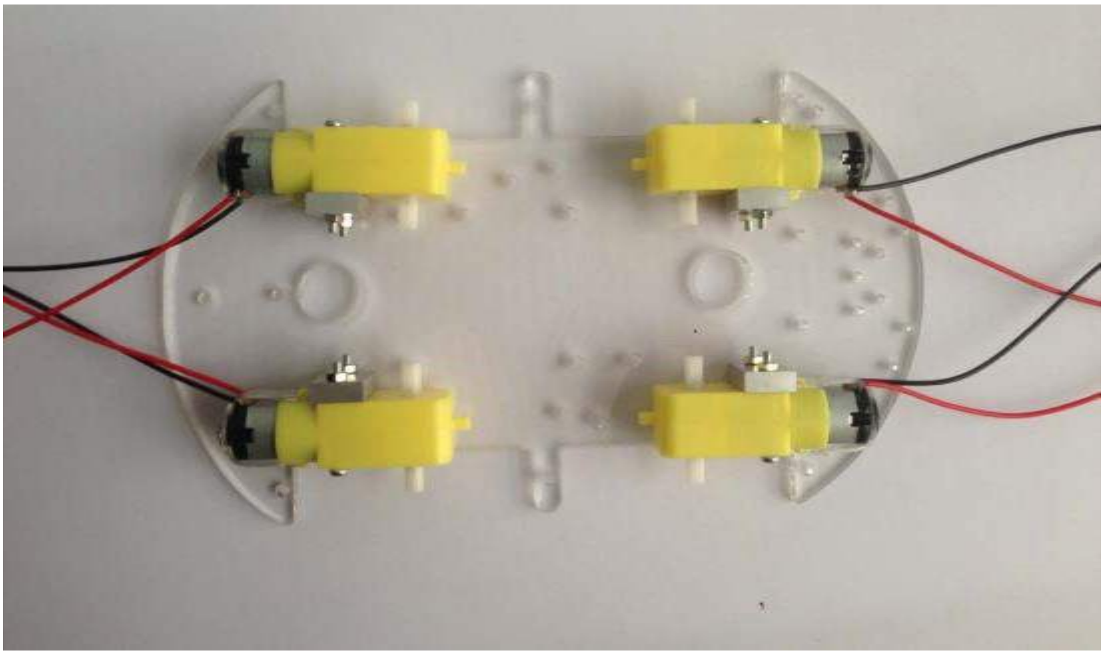


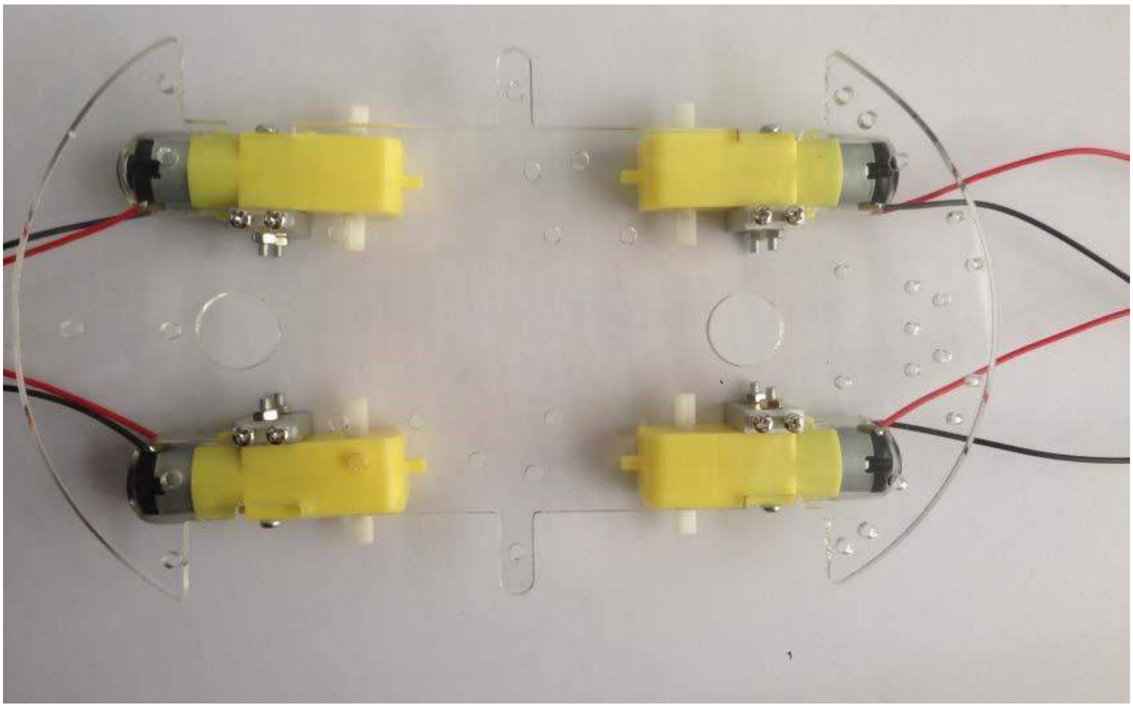


After installing the motor, the motor can be fixed into the plate.

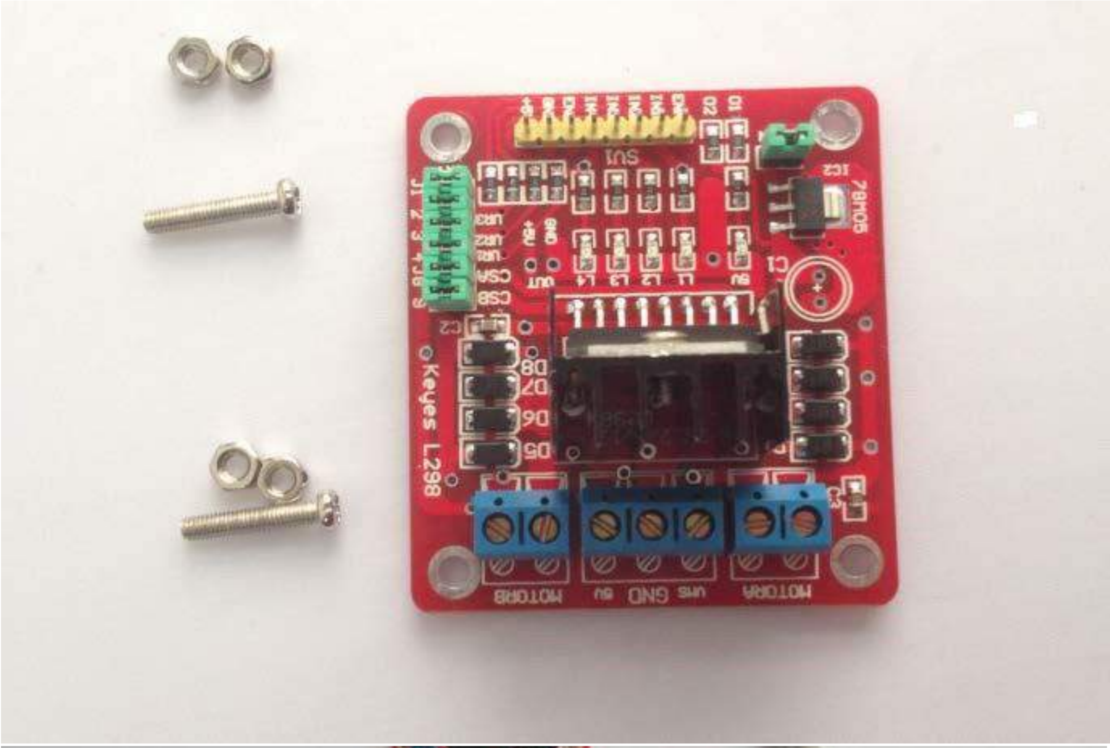


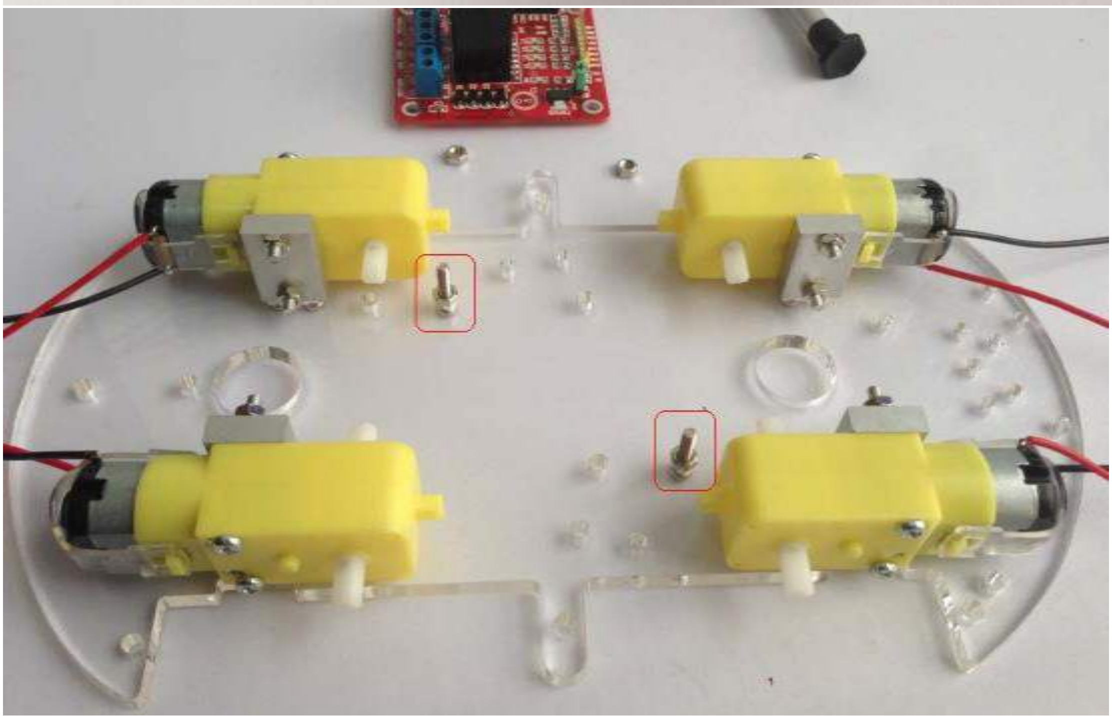




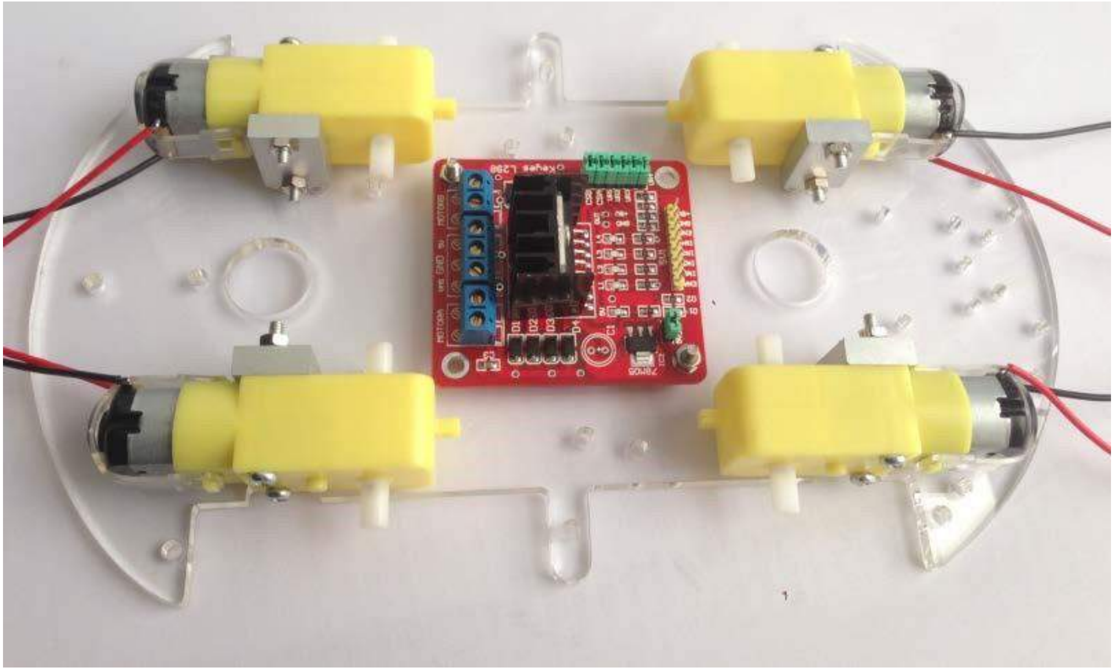


Here we fixed the motor, next we are going to fix the motor drive board.

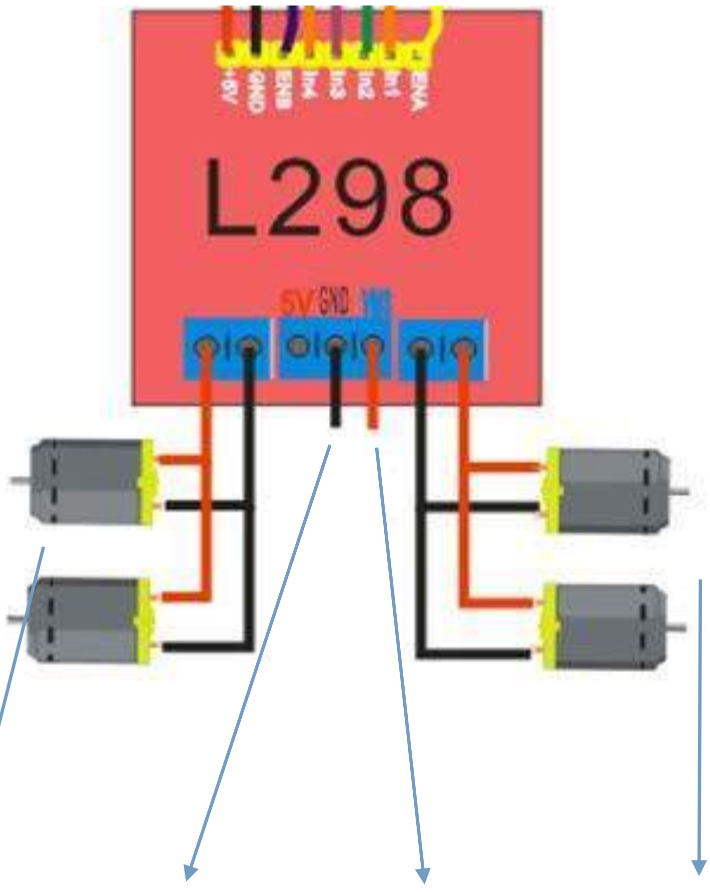




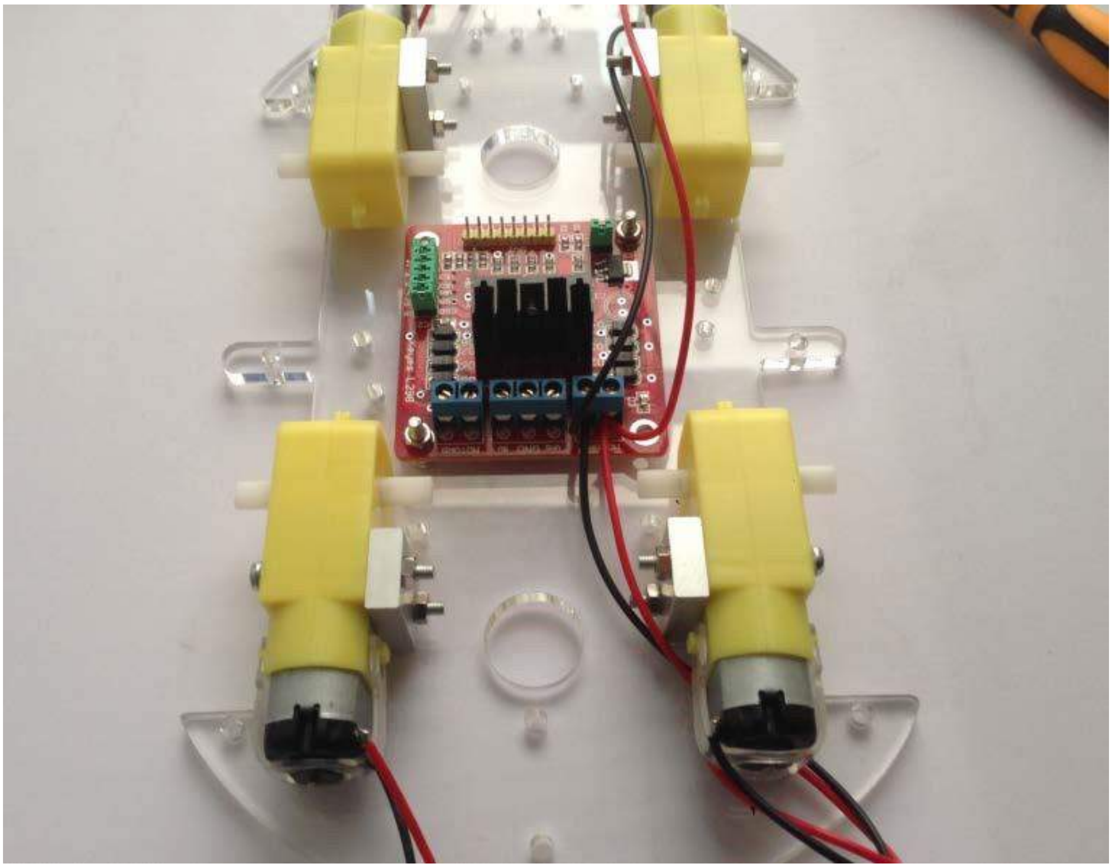
First fix up two screws with nuts that match onto the plate for mounting the the motor drive board.



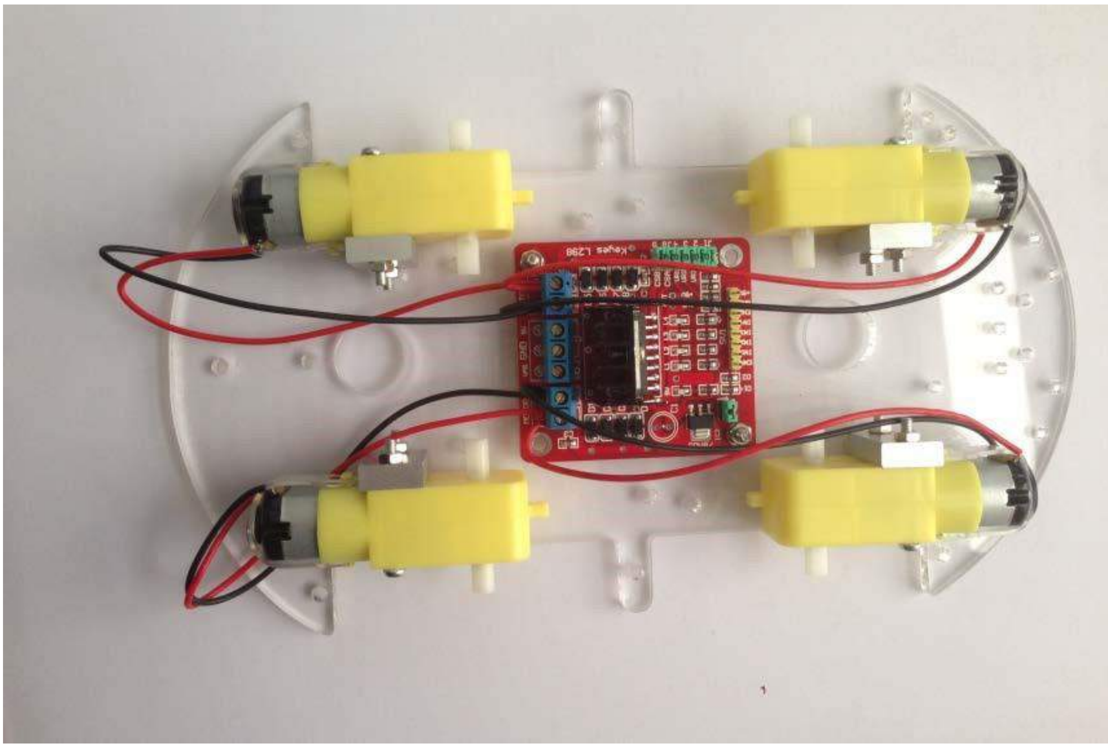
Connect the motor cable, Circuit diagram:



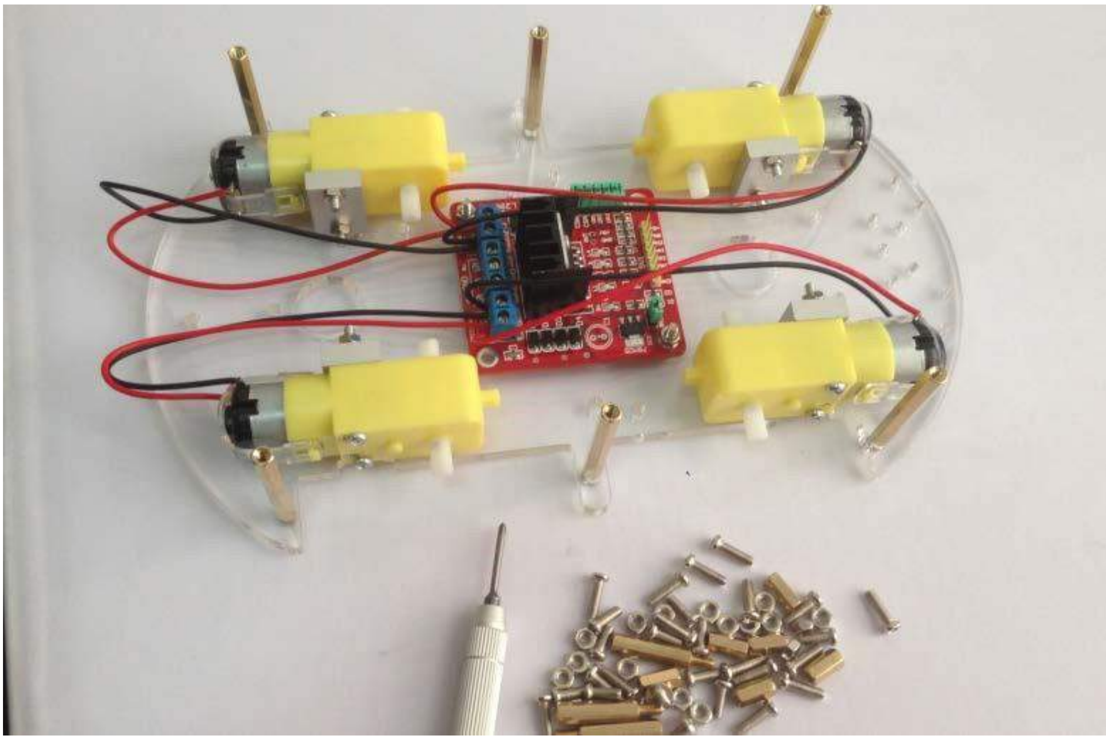
Left wheel Negative Positive right wheel



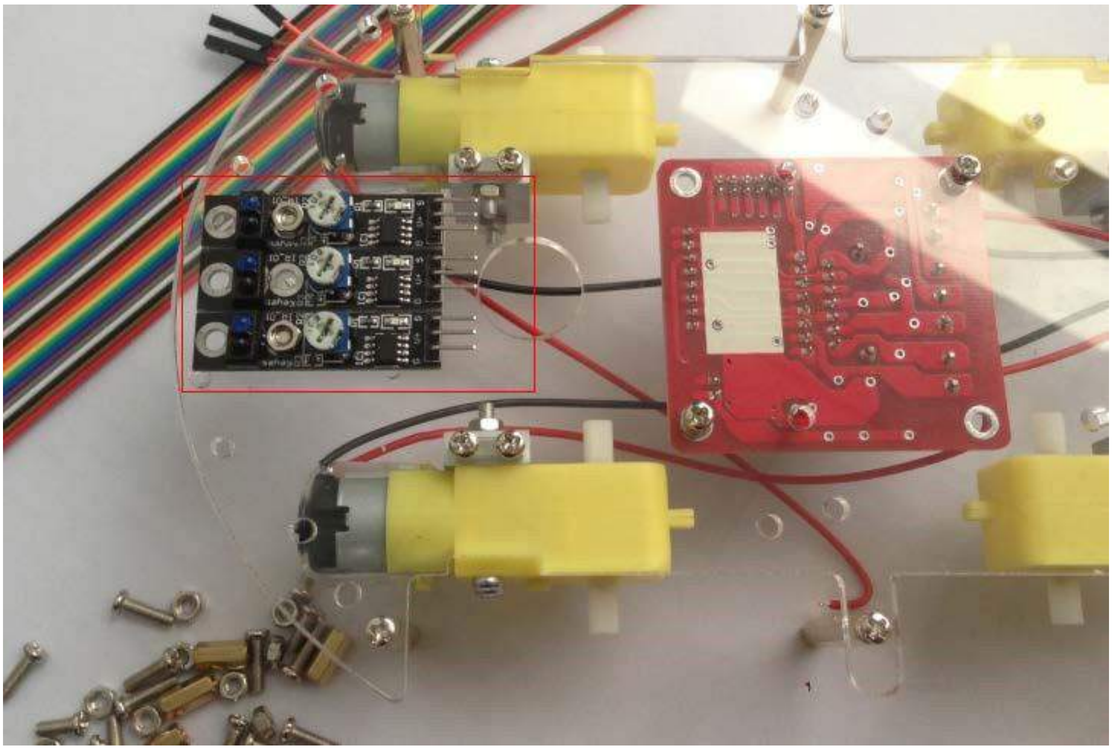
The left front and rear wheel as a group, the right front and rear wheel as a group.



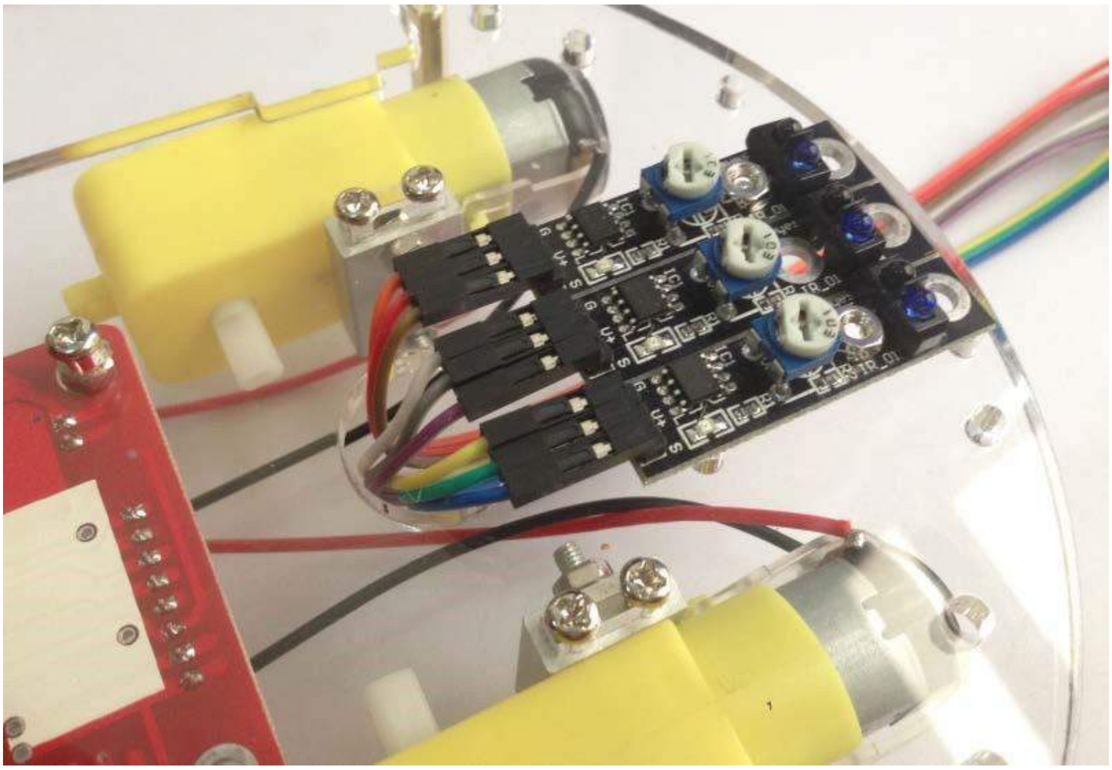
Fix on the long copper column：



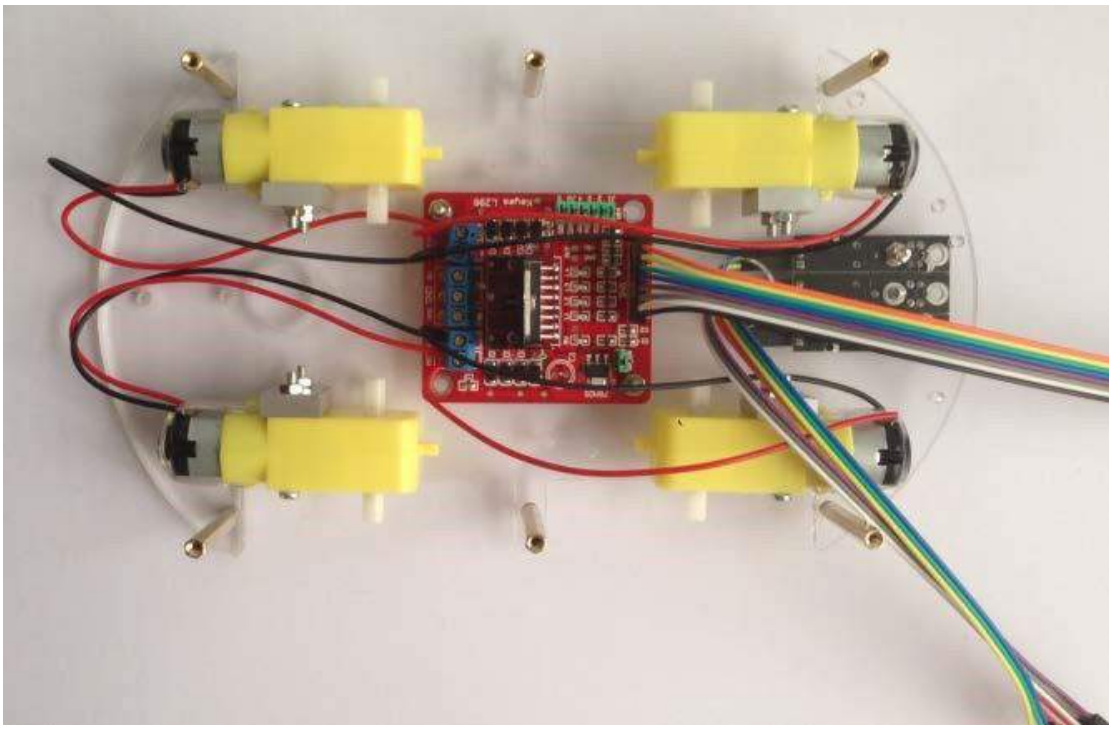
Install the tracking module under the chassis front and fixed it by a screw



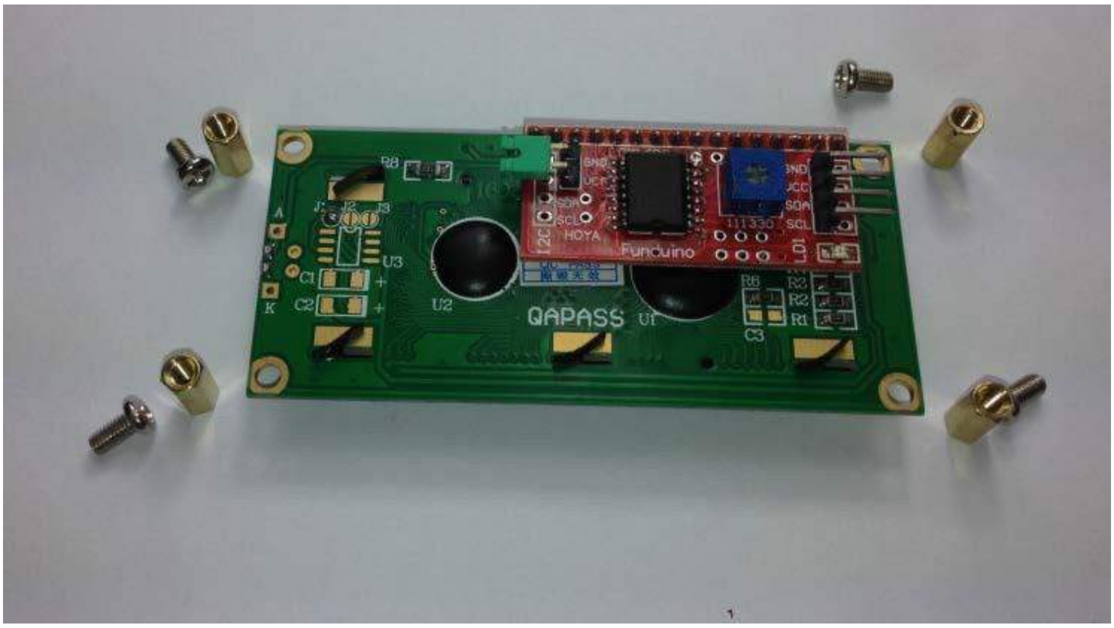
Plug the DuPont line in order and led to the top through hole

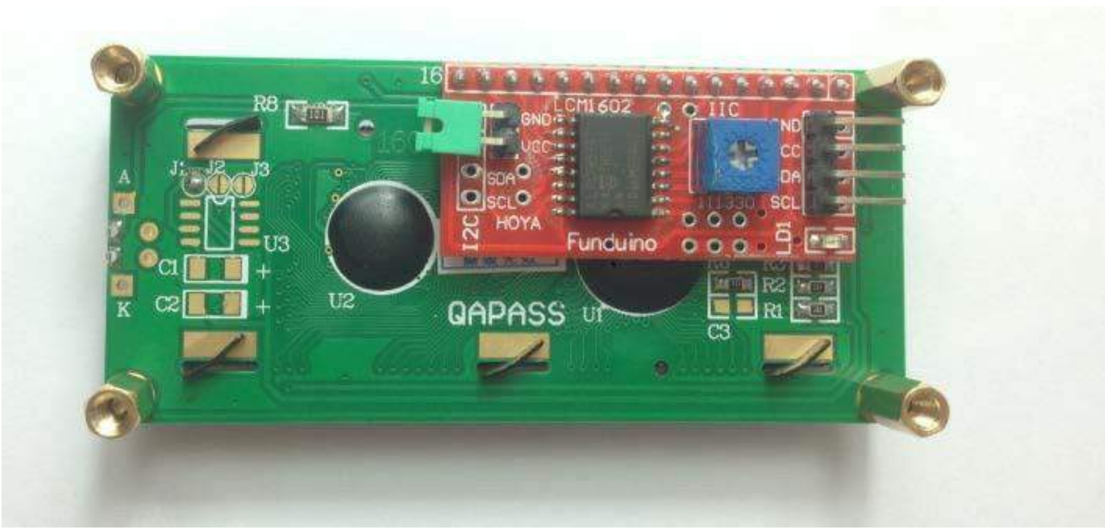


Plug the DuPont line of motor driver board in order, don't mess them.

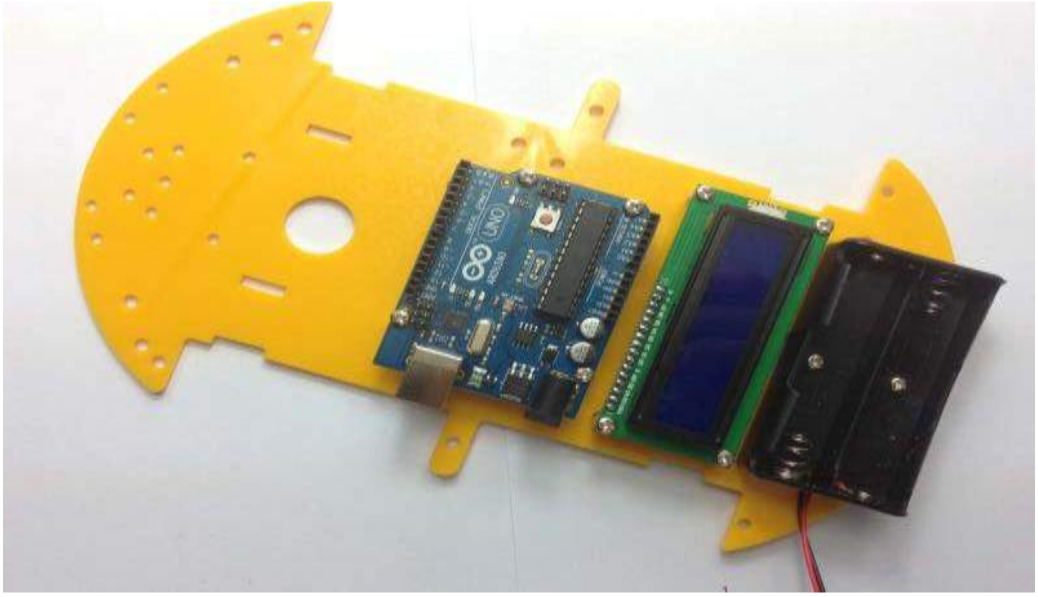


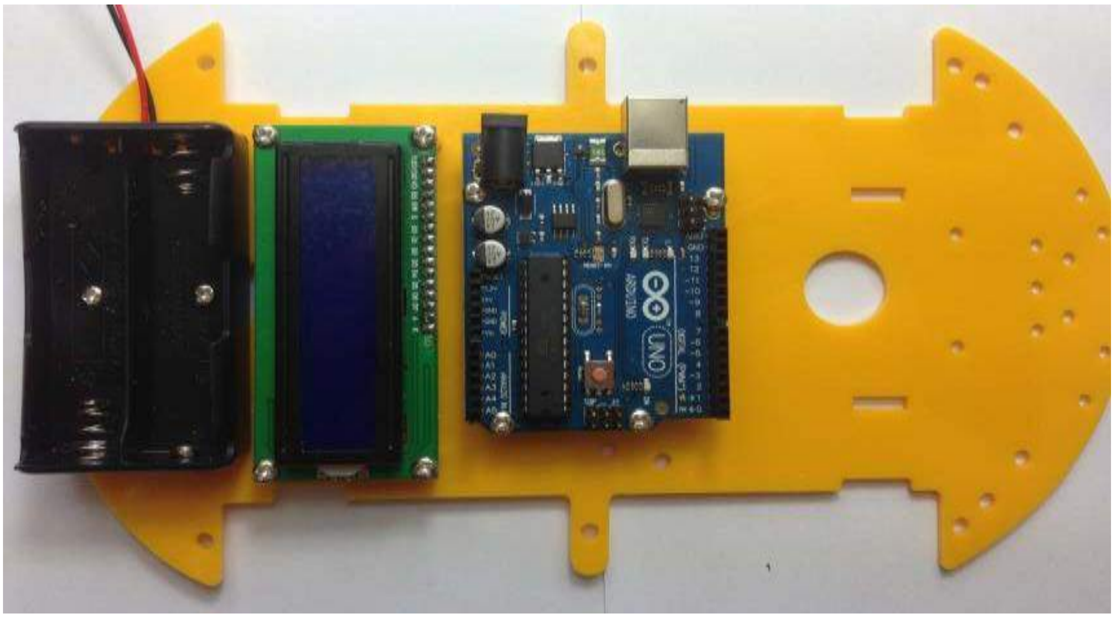
The I2C interface of LCD1602 LCD, fix it with short copper column：





The I2C interface of LCD1602 LCD, and the Arduino board, the battery box are both fixed on the yellow board.

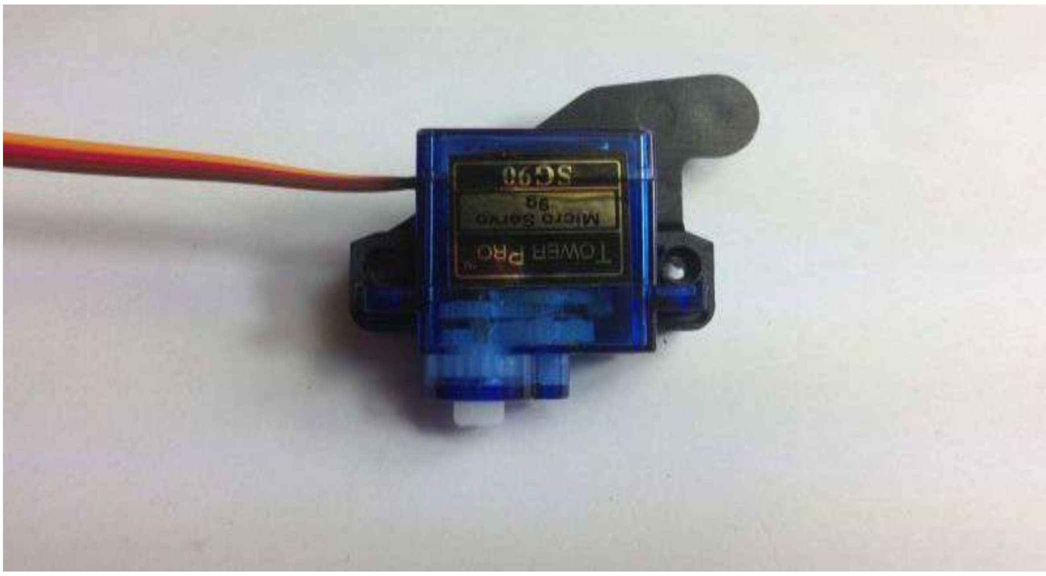




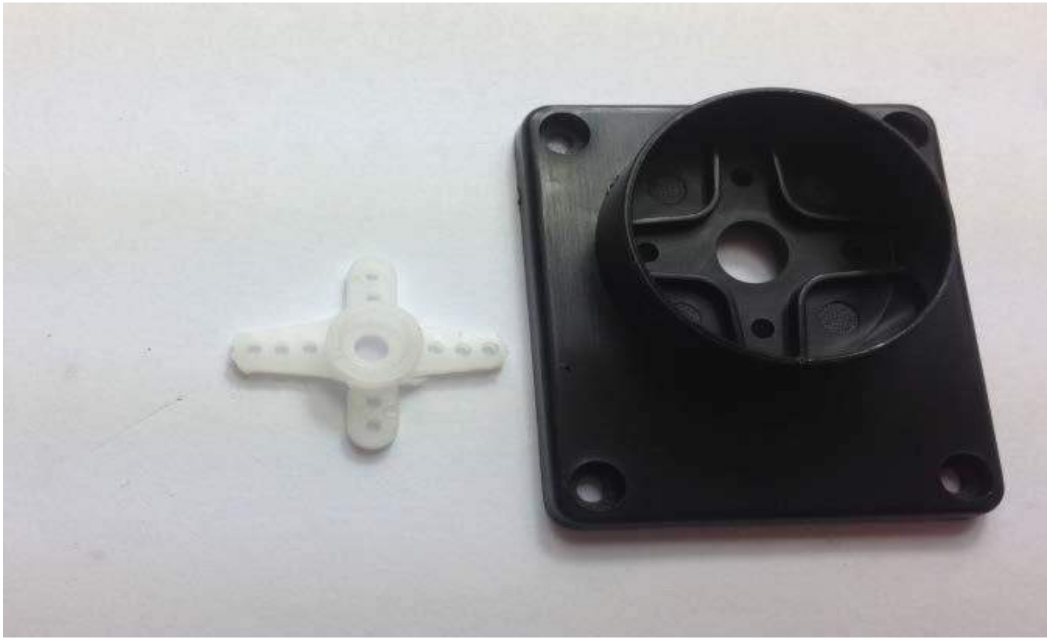
Next, we begin assembling the holder, and the ultrasonic module：

We will use the servo, holder set, and untrasonic module：

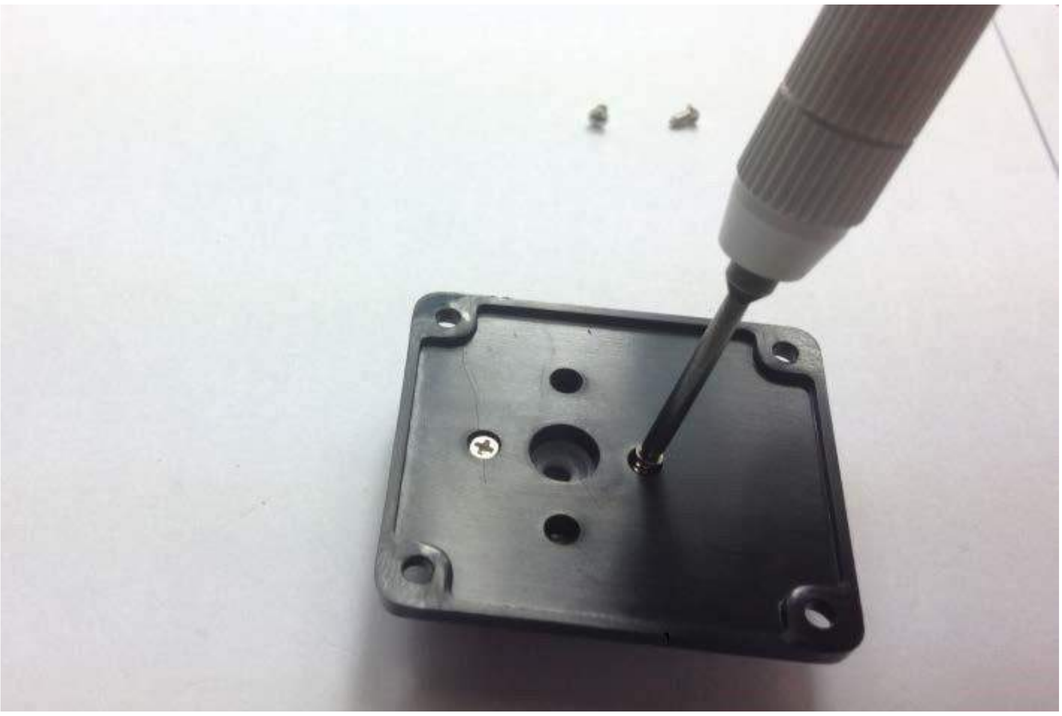



SG90
Micro Servo
9g
TOWER PRO





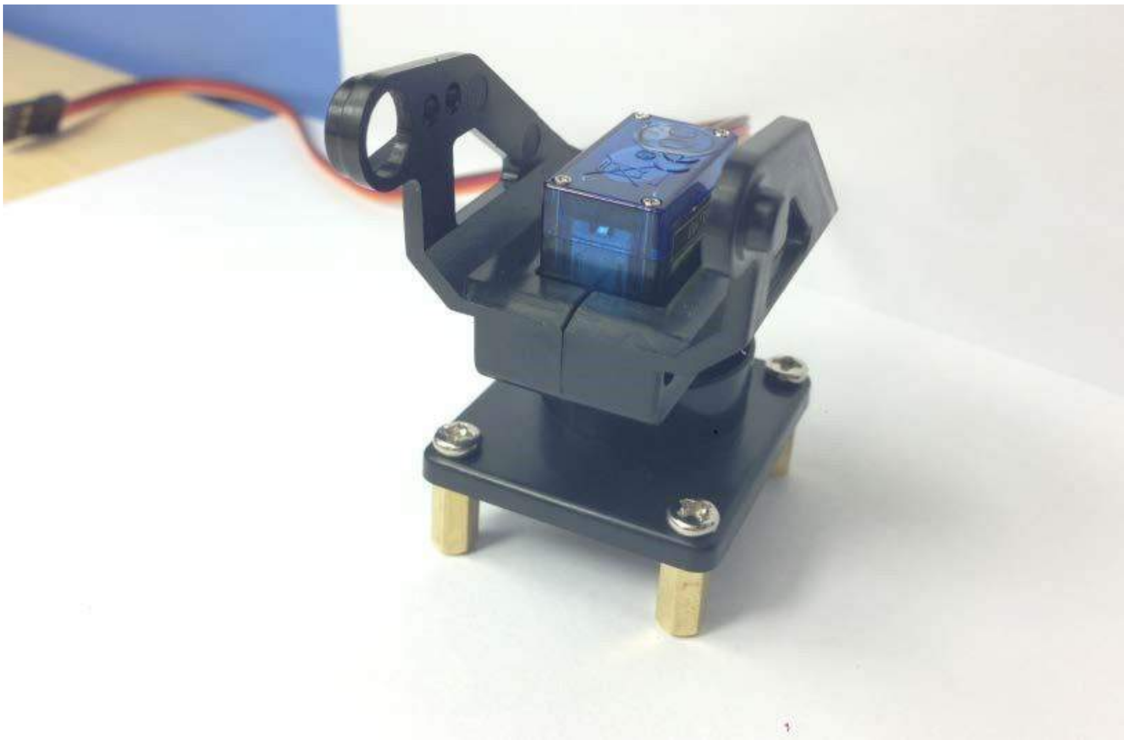
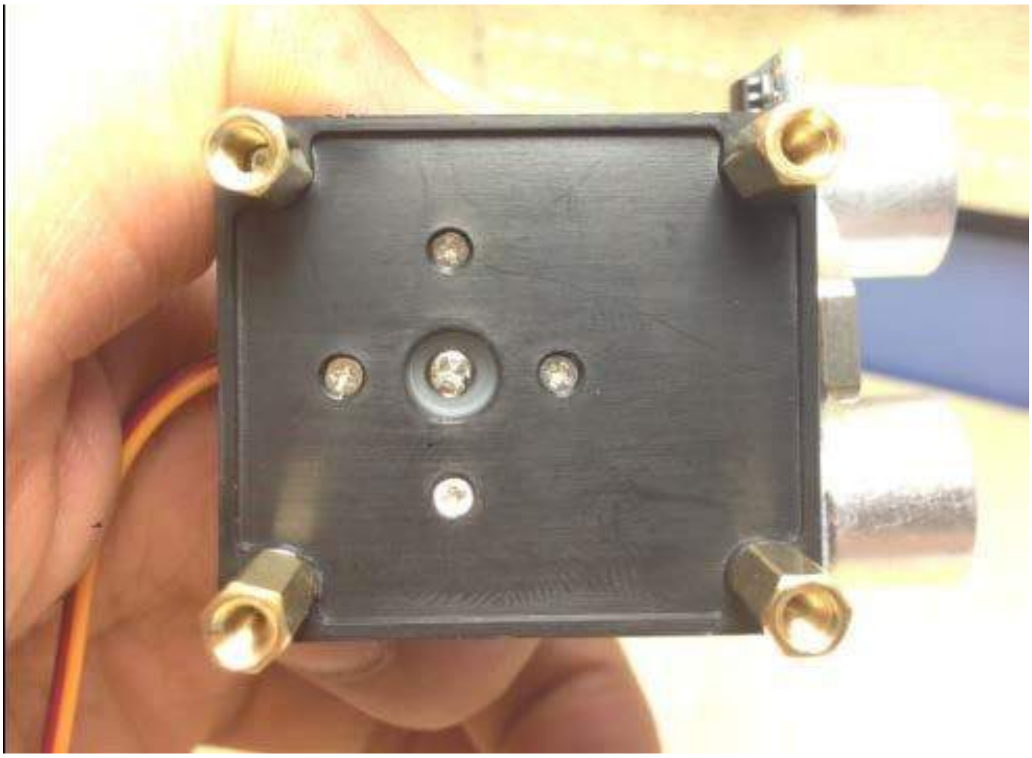


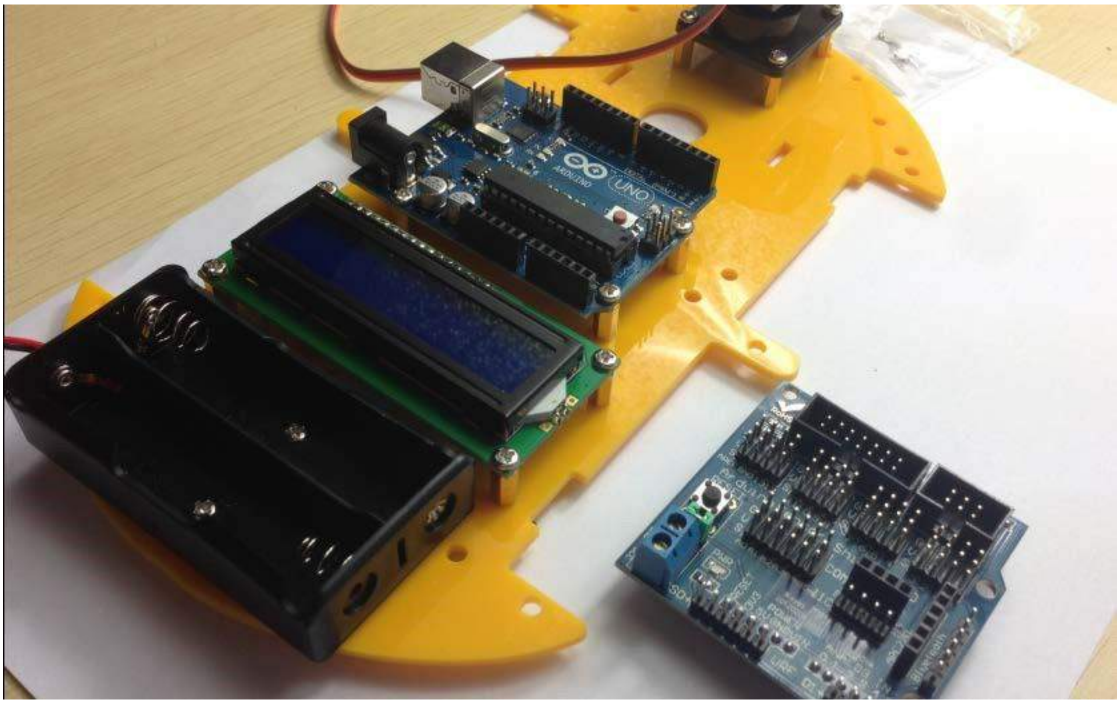


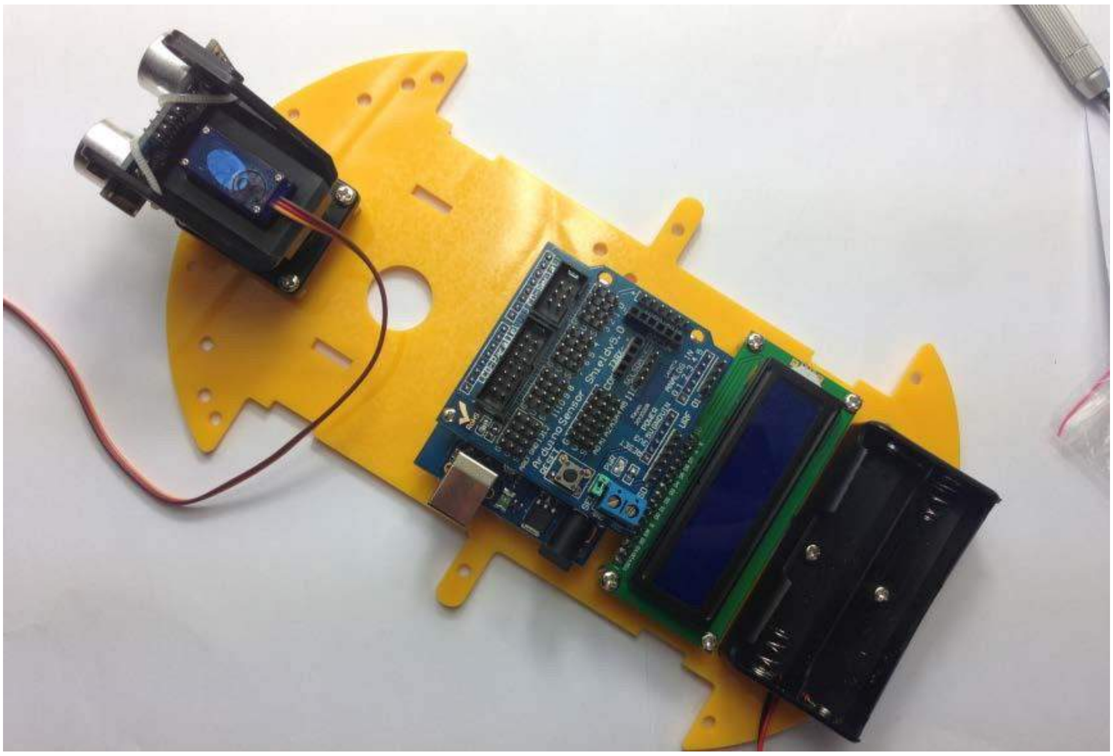


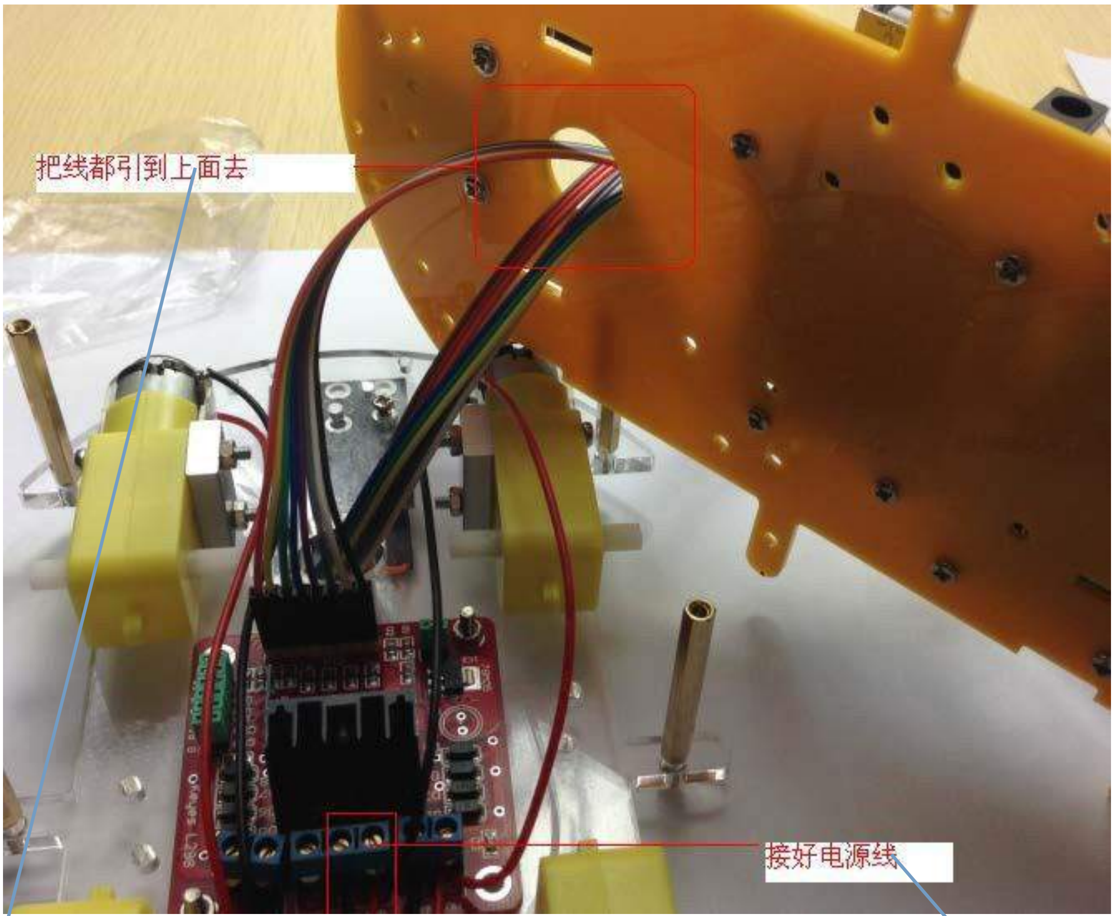


After finished above, sway the servo from side to side, let the servo face in the front in 90 degree. Install the copper column and fix up the middle screw.



Then, insert the Arduino compatible board,fix the servo motor holder on the head of smart car.



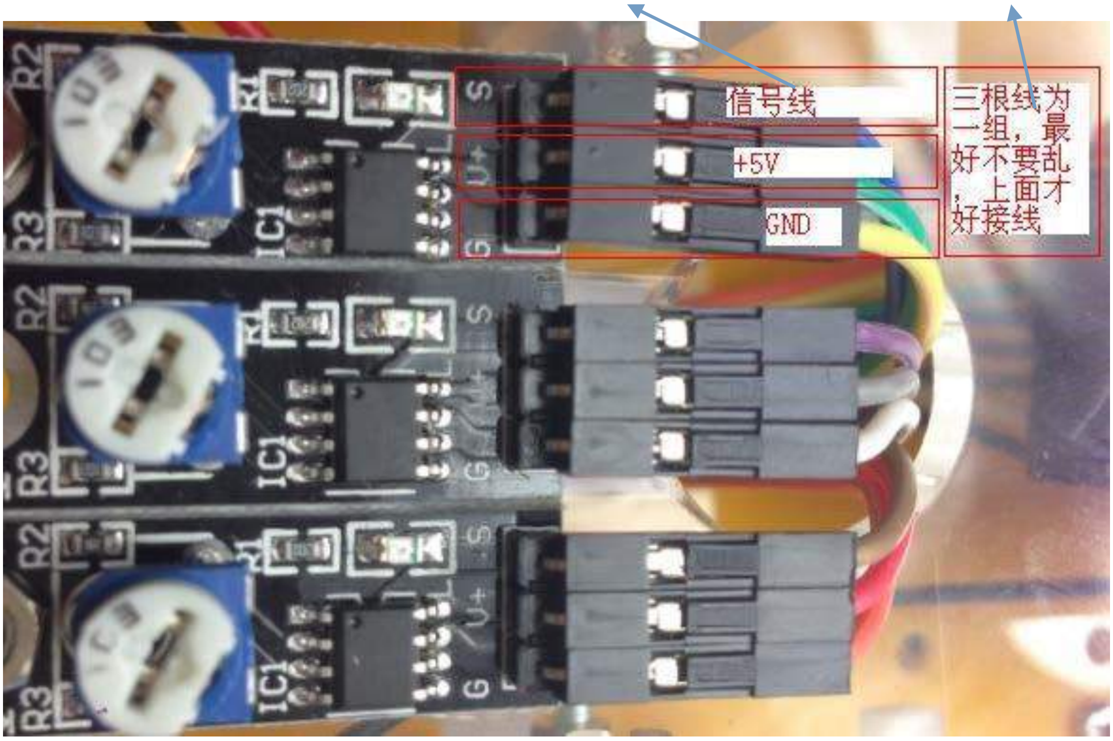


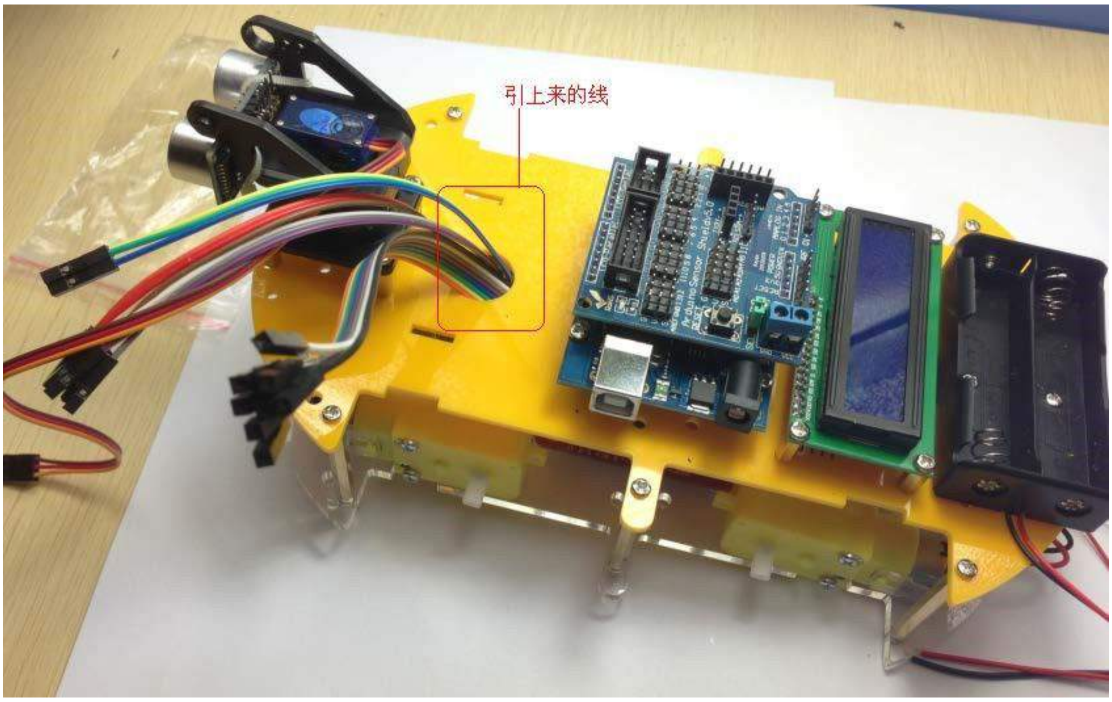


Lead the cables to the top connect the power wire

Three lines as a group.don’t mess.

Signal wire

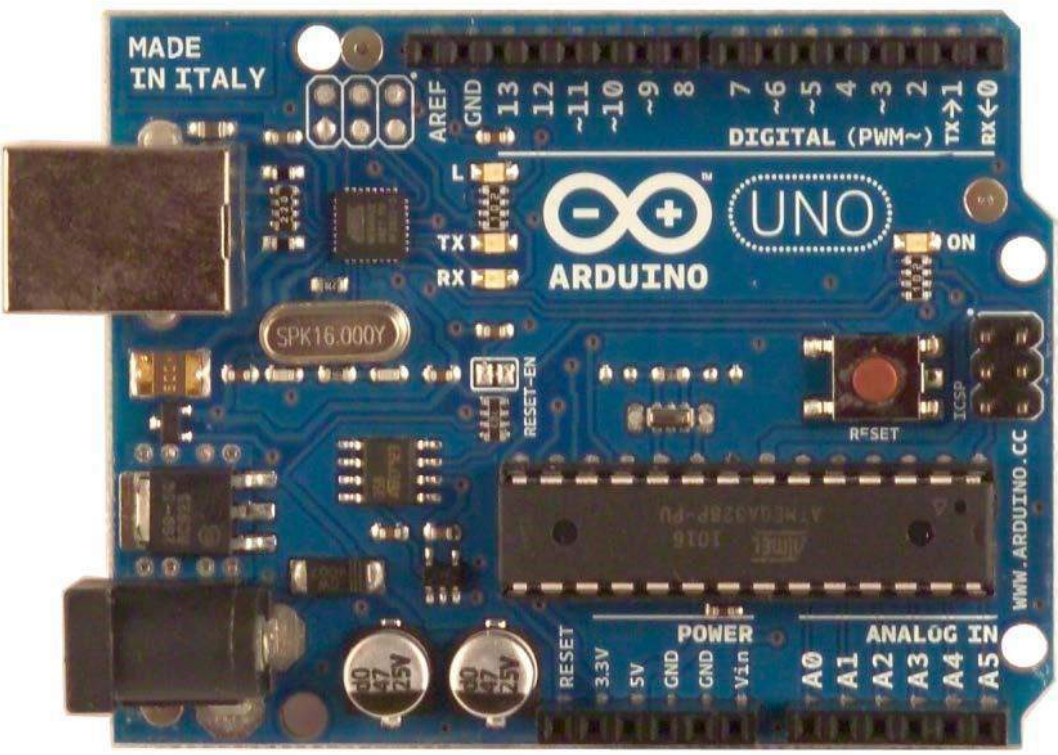




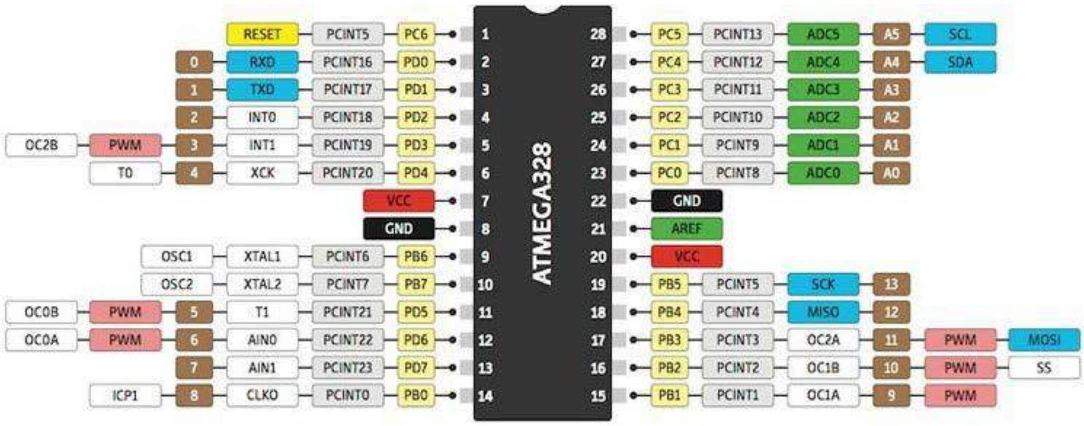
We finish assembling the vehicle bottom, and now let’s start testing the module

6. The Use of Arduino MCU Board

1. Introduction of Arduino



Arduino derives from an open source hardware platform project in Italy, the platform includes a I/O function with a simple circuit board and a set of application software development environment. Arduino can be used to develop interactive products, such as it can read a large number of switch and the sensor signal, and can control the lights, motors and other every kind of physical device; Arduino can also develop a peripheral device connected to the PC, can at run time with the PC software for communication. Arduino hardware circuit can be welded together, also can buy already assembled module, and program development environment software from the Internet can be free to download and use.



Let us see how the Arduino team define it：

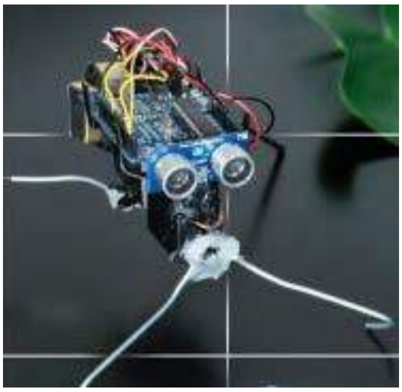
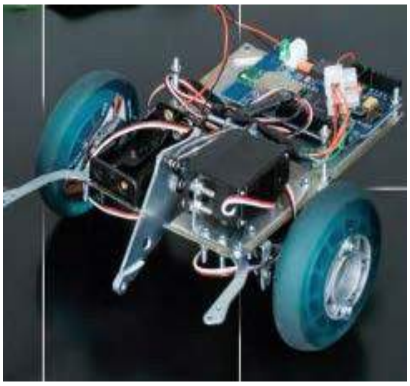
Arduino is an open source electronic prototyping platform, have a flexible, easy to use hardware and software. Arduino is designed for designers, arts and crafts, amateur, and interactive device for the development of interactive or development environment for people with interest.

Arduino can receive input signals from various sensors to detect the running environment, and through the control of the light source, electric and other drive to influence the surrounding environment. Microcontroller programming using Arduino programming language board (based on Wiring) and the Arduino development environment (based on Processing). Arduino can run independently, can also be run with computer software (e.g., Flash, Processing, MaxMSP) for communication.

Arduino hardware circuit board can be welded and assembled, can also purchase has been assembled. The software is available from the Arduino web site for free download. You can obtain hardware reference design (CAD file) according to the open source license to, and the freedom to modify to suit your needs.

The definition of Arduino is still a little fuzzy, this is where the advantages of Arduino. Arduino is the glue people to connect various tasks. To give the Arduino a most accurate definition, it is best to use some examples to describe.

You want to be Coffee for HERSHEY'S, Coffee pot issued a "squeak" sound to remind you?You want to be a new mailbox mail, phone alerts to notify you?Want a shiny toy?Want a distribution function of speech and drinks professor X steam punk style wheelchair?Want a press can be tested buzzer?Want to be your son made a "warrior" the Milky Way arm cannon?Want to make a heart rate monitor, each bike records stored in the memory card?Thought made a drawing on the ground, can ride in the snow in the robot?

Arduino can achieve all of these.

Don't understand to electronic products or micro controller of the people, this sounds very cool, very interesting, and will want to join the club. This is the children wanted to make things, you can even put them in the process of high school knowledge. These projects have in a science fiction story, these small devices in the log. The common ground is the fantasy, is some dream of things. But now, these ideas are really implemented, timely you not an engineer, but also can be made.

This is a big deal, because the engineers often as other engineers to design the development platform, and not for the artist, weirdo, or to share an idea and simply connect the things children. The Arduino team is not composed of "hardcore" electronic engineer, but by the designer, teacher, artist, and all "hippie" I know composition (hippie is a compliment here I hope I didn't offend them). Arduino's main base in Italy, every year I will see the articles about the Italians trying to find "Google belongs to me", in fact they already have, that is Arduino, but they have not yet realized this point.

Have a look the instance of Arduino project, you will find the makers for these electronic products "is what" more interested in, rather than making method. These Arduino fancier often said Arduino is not taught in basic electronics, "bah, this is what the electronic products really ah," they say, "too easy!" Yes, indeed. If you are an artist or designer, without the use of a Arduino case, want flashing light-emitting diodes, or motor rotation, so I wish you good luck. Of course, if you are willing to spend money and take your heavy electronics technology teaching materials to a show, this is not a bad idea. But for other people there, they just want to use the light emitting two tubes to decorate the burning man festival clothing.

For some traditional microcontroller Arduino community is how to treat this problem, I think the best example is AVR Freaks, the official website focused on AVR processor (also used for Arduino). Would you like to AVR Freaks the community would love Arduino, because Arduino can be AVR micro controller to the masses. However, many people do not like the site all the weird stuff these non engineers and manufacturing, as these will destroy their hierarchy. I most love quote (I hope this sentence on a T-shirt).

"Arduino programming language, the child can understand, use once and addiction" -- ArnoldB, AVRfreaks website

In fact, the wrong attitude is to promote Arduino fans to create your own community, help Arduino build a swallow anything and everything, but not superior community.

Arduino is a simple, but not too simple. It is around these ideas to establish, in which students use Arduino to realize the purpose of: receiving the sensor signal, get some code, and then use these signals and code. Perhaps even can't write code, they cut and paste the code after the beginning of. Arduino is the thermal binder, rather than precise welding. Therefore no one will be cut off a hand, also won't have laboratory burned. While the Arduino team member professor of Arts and crafts workers and designers how to use. Every day, Arduino in the study, Professor, the establishment and continuous improvement and code sharing project. These arts and crafts workers and designers are on the Macs system using Processing language and modify. (the Processing is Arduino's brother)

Said here, Arduino is like a warm, no boundaries, assembly of strong art atmosphere. This is the Arduino become a "do it yourself" why successful model? Not only that, we have to find more specific information.

Library -- a simple task, complex task easily.

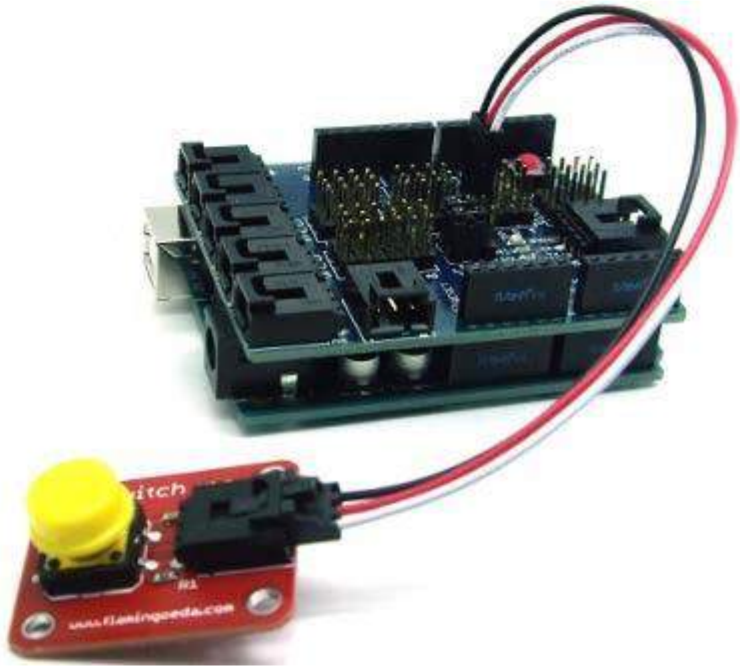
Package library was widely used to perform complex tasks, such as writing the SD memory card, write the LCD driver, GPS. There are also some libraries used to do simple things, such as rotating pin or button debounce. If there are 10 chips, we have to install UART code to write 10 times, frankly, this boring. If the call Serial.begin (9600) function to handle the register data, it will be much easier for you.

Light weight, run directly on the hardware level.

The use of a certified, easy to understand the compiler (we can even say that AVR-GCC is the default AVR or standard compiler), the code can be run directly on the hardware level. The compilation and NET language and BASIC language, the compiler runs fast, small volume, light weight, and you can use the HEX (sixteen m) file for the new chip batch programming.

Sensor

The real reason for soaring Arduino is capable of realizing the analog input into a digital input, in other words, you can use the light, temperature, sound, or already on the market any low cost sensor signal input, Arduino can identify. For the digital sensor, Arduino support SPI (high speed synchronous serial port) and I2C bus. The functional coverage sensor market 99%. The use of other development platform is not easy -- imagine if a piece of Beagleboard (great product) and Arduino are tied together, just for the sensor data, which is really strange!



Simple, but not too simple

The traditional development board is often too complex. There are many accessories, such as liquid crystal display, button, emitting two tubes, 7 digital tubes etc. Development board shows all its functions. The number of display board Arduino function is the absolute minimum, if you want to achieve functional expansion, only need to add Shield (shield). there are hundreds of thousands Arduino Shield, from the liquid crystal display to the wireless Internet technology, but to increase the number of Shield by users themselves. Extend the functionality of Shield is also easy, for the production of Shield extensions will be commercially stimulation.

Non chip manufacturer

Arduino development board is not designed by the chip manufacturer. Why stressed this point? Because the chip manufacturer to highlight their products out of the ordinary, they will often add some strange things. Arduino stressed micro control common rather than differences between. This means that Arduino is a good beginners platform, as long as the Arduino board can do, you can do in any other micro controller. This feature will accompany you for a long time.

2. Installation and application of Arduino driven programming

Frist download Arduino development software，web address：

http://arduino.cc/en/Main/SoftwareDownload an arduino-0023.zip，unzip it to the computer hard disk.

Arduino UNO will connect to the Windows through the USB line, will appear "Arduino UNO", the new USB devices is found.



Then Windows will lead us into the "found new hardware wizard" window, select "no" option and click "next" button:



The next step is required to install the Arduino UNO drive, select "install from a list or specific location (Senior)" option and click "next" button



Put the Arduino UNO USB driver on drivers directory which is in the Arduino 0021 installation directory, we need to specify for the windows that the directory is the search directory when installing driver:



Click the "next" button, Windows begin to find and install the Arduino UNO USB driver:



If all goes well, we will see the following success interface:

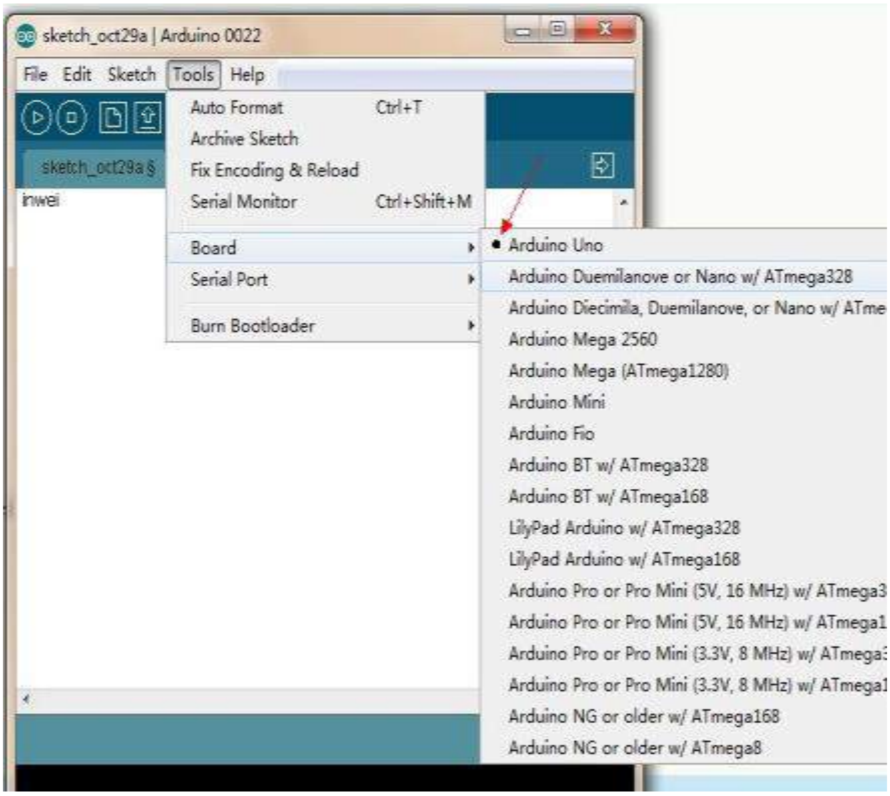


After the Arduino UNO USB driver installation is successful, we can find the corresponding Arduino UNO serial port in the Windows device manager:

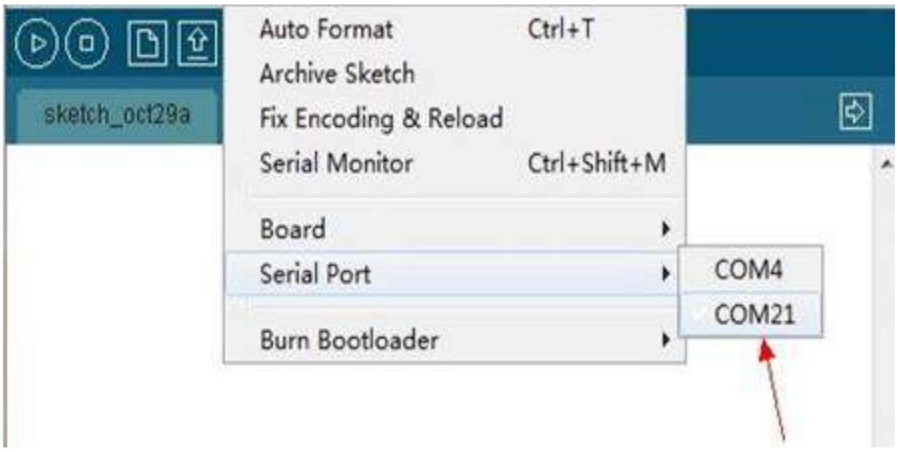


Let’s writing the first program for example，light up the “L” light

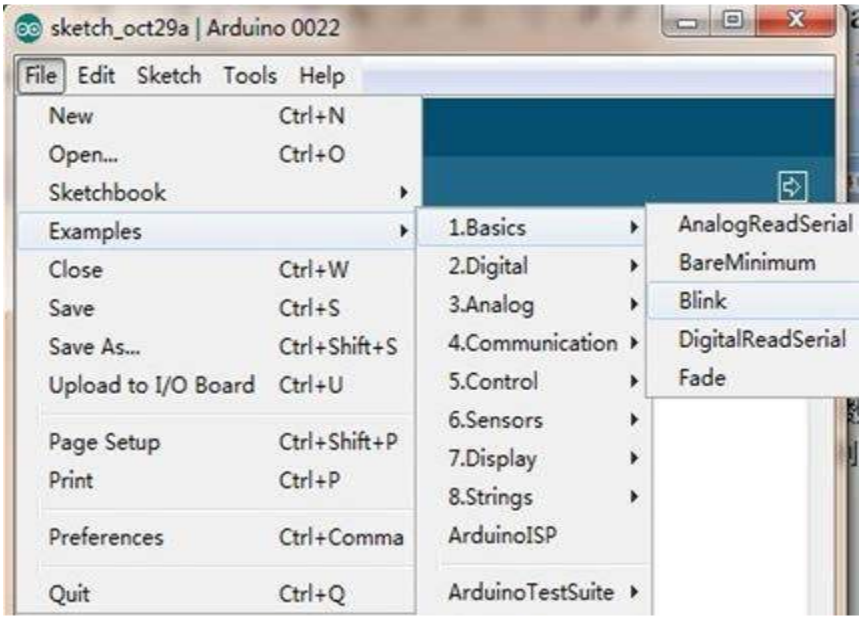
In the Arduino-0023 programming interface, click on the [Tools], move the mouse down to the menu [Board] option, see continually pop up menu, in front of [Arduino UNO] whether there is a black spot, if not, then click the [Arduino UNO] this option.



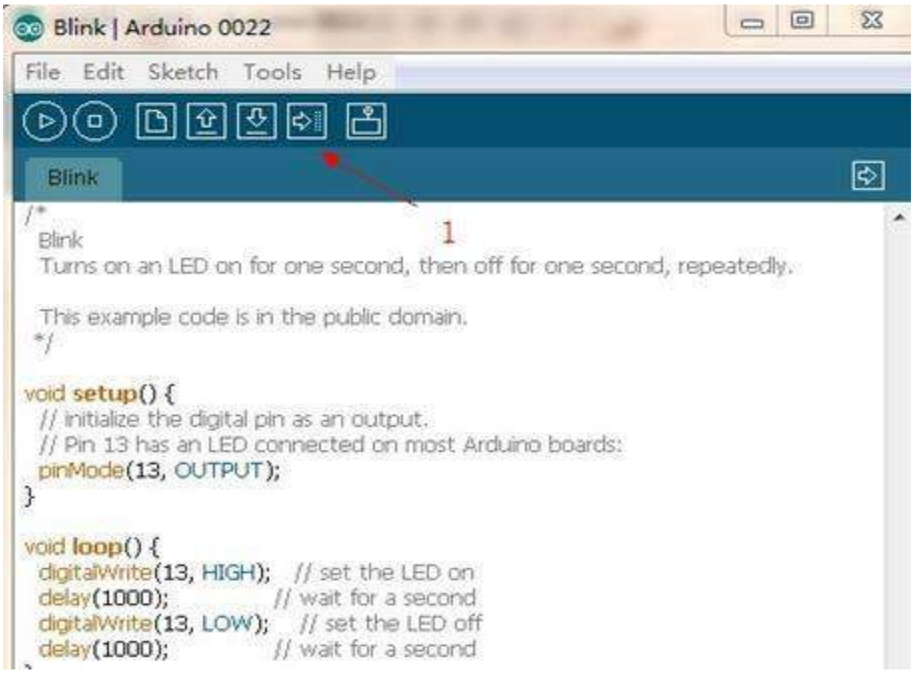
Then, select the right COM. Remember when you installed the hardware requiring you to record (COMX) value: X? Here it will be used. If you installed Arduino UNO port is 21, so the mouse clicks 21



Let’s import a sample program to flash the “L” light, mouse clicks【File】 click【Examples】in the pop-up drop-down menu， extend to【1.Basics】menu, find【Blink】， and click【Blink】



After click the 【Blink】, there pops-up a Arduino programming interface.



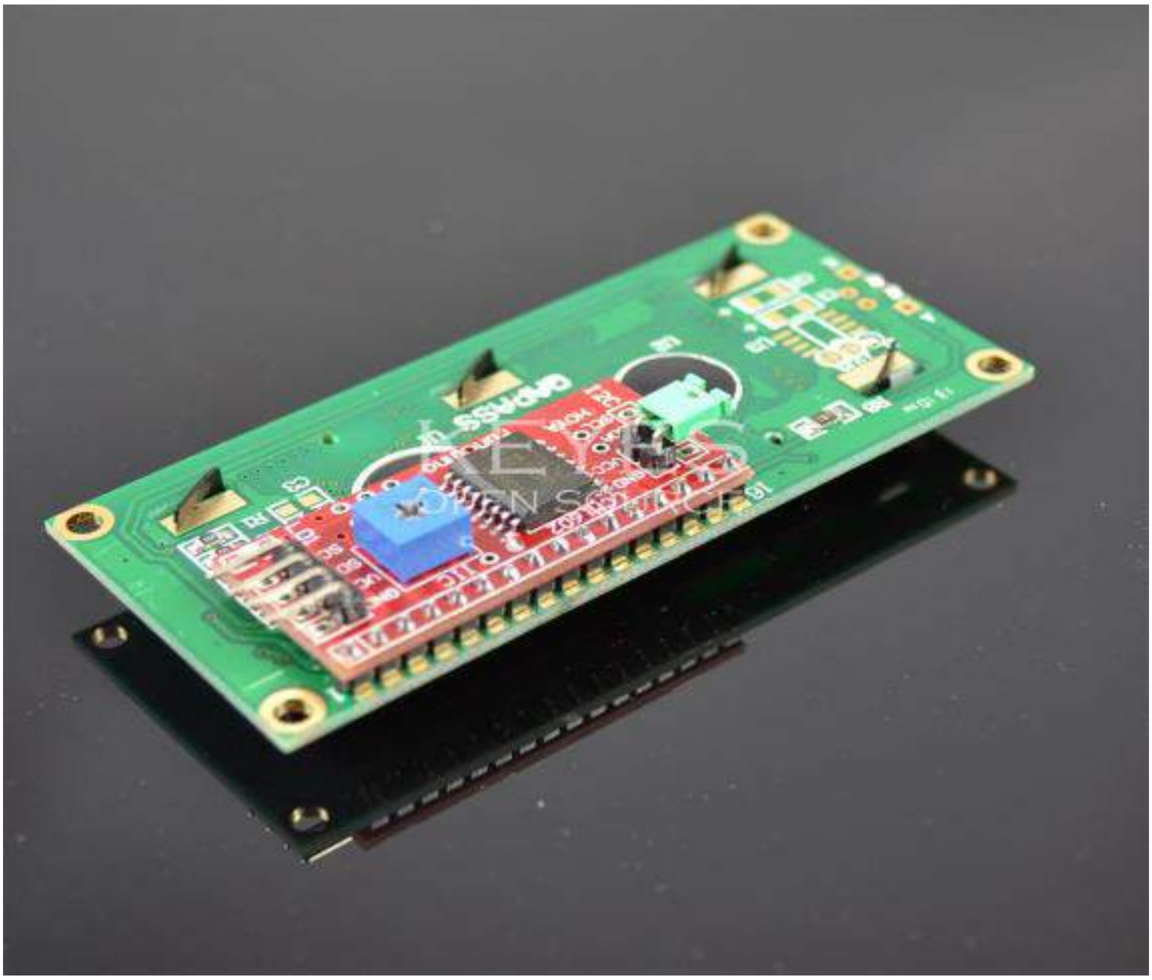
Directly click the icon that the red arrow 1 refer to, you can find that there are two yellow lights on the Arduino board flash for a while. By the two crazily flash yellow lights put off, it appears words hints under Programming frame: 

Light “L” on and off one time a second on the mian board, and congratulations, your have done your first program!

7. Experimental Details

1. I2C LCD1602 Module Instruction.

Instructions before Use：Put The Liquid Crystal_I2C Library folder into the Arduino libraries directory.



Arduino board just has 20 IO ports.

Original 1602 display needs 7 IOports to drive it, but this module can save 5 IO ports for you. We can send you the Arduino library.

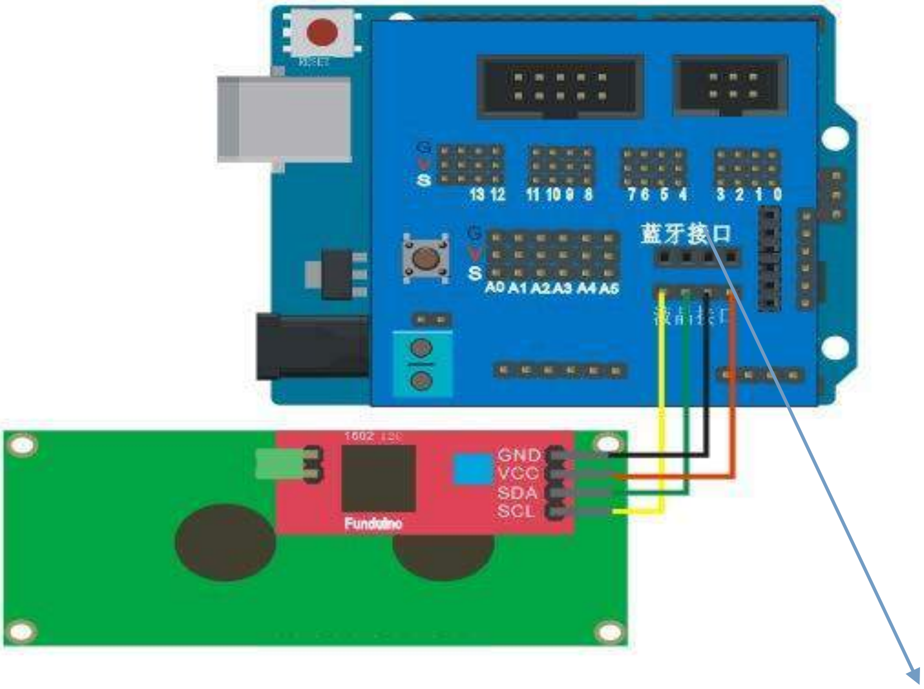
Parameters:

1.supply voltage：+5V

2. Support I2C protocol

3. A backlight, and contrast adjustment potentiometer

Below is the connection circuit of module and the car

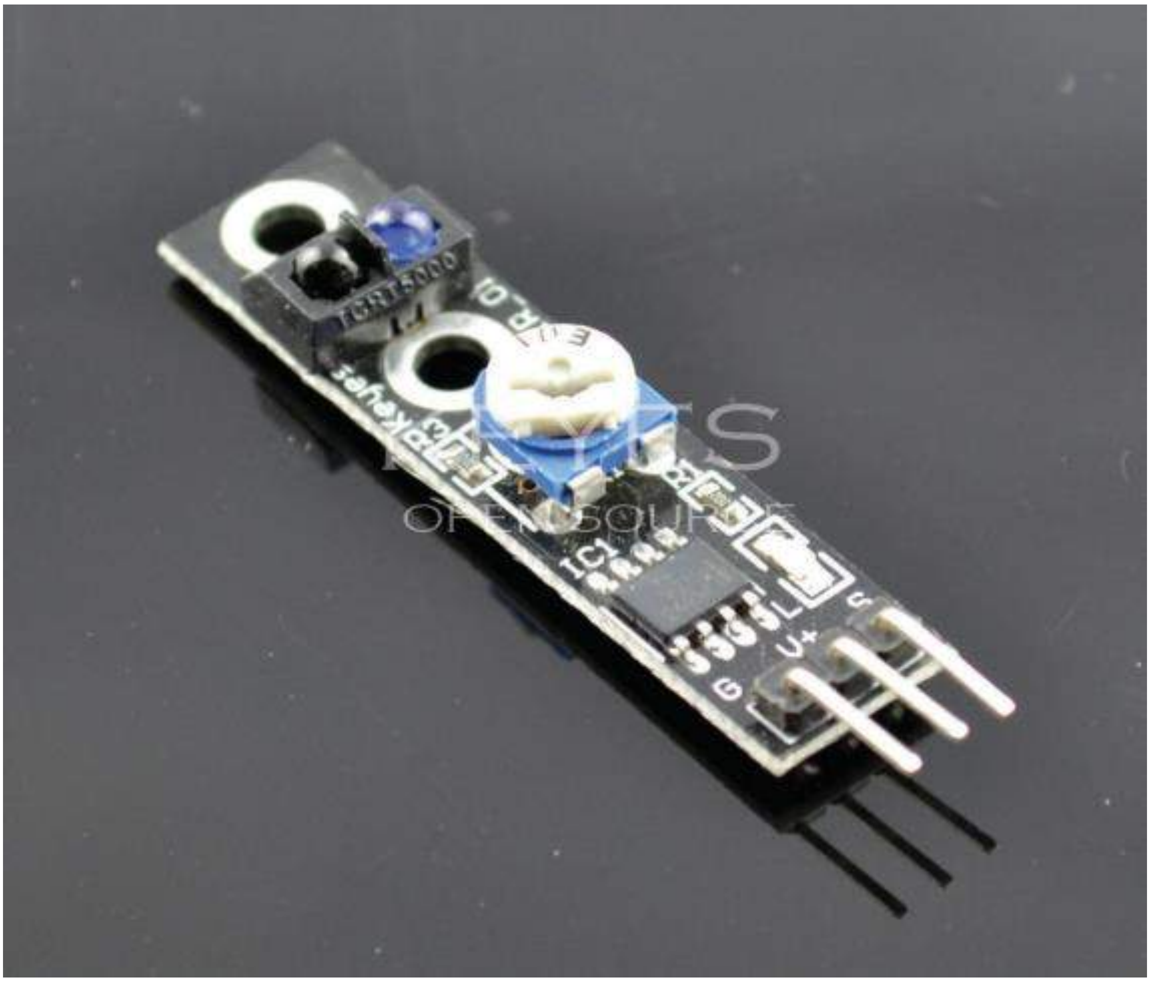


Connect the wire, provide uploaded the code to Arduino, can display characters.

```
*****************************************************************************
**********

#include <LiquidCrystal_I2C.h>    //包含液晶的库
#include <Wire.h>                 // 包含 I2C 的库
LiquidCrystal_I2C lcd(0x27,16,2); // 定义了一个 1602 液晶
void setup() {
  lcd.init();                  // 初始化液晶   lcd.backlight();              //
开背光   lcd.print("Arduino");          //显示 Arduino
} void
loop() { }
*****************************************************************************
```

2. Tracking Module Instruction for Use

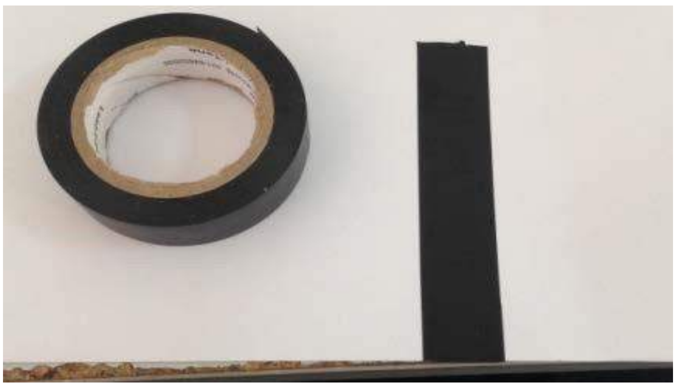


The working principle of module：

When the black line is found, the sensor output is high, the output is low when don't find the black line.

The adjustable resistor on Module is used to adjust the sensitivity, adjust to the appropriate location, here to teach you how to adjust.

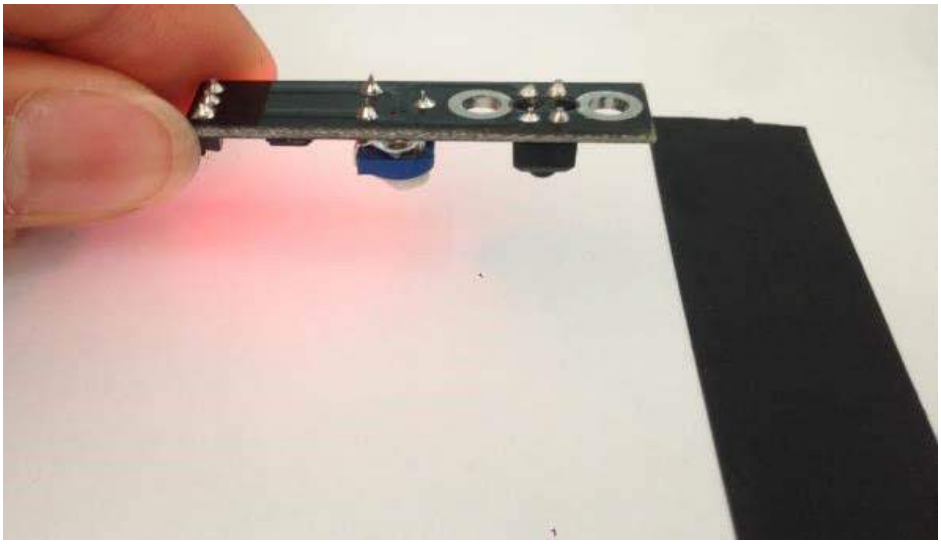
Find a piece of white paper, with black tape.



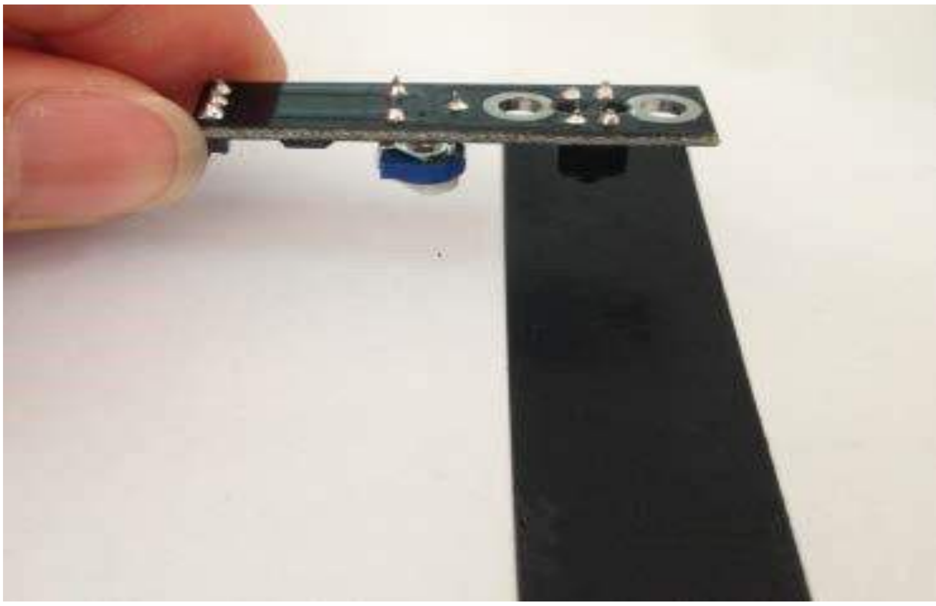
Connect the tracking module to the power supply only by VCC and GND.

Put the module above the paper about 1CM, swing it between white and black, watching if the red light on the module has a change, if not change, adjust the variable resistor.

When the module doesn't encounter the black line, red light is on.



After encounter the black line，LED put off

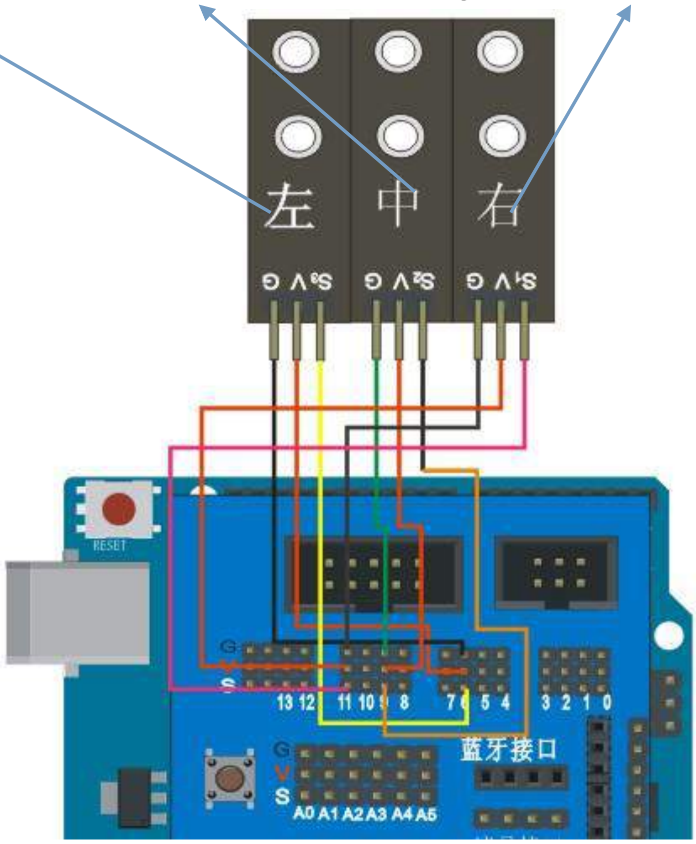


So the module is well regulated, itcan be fitted to the car.

Below is the connection circuit of module and the board:

Pin corresponding relationship

Left Sensor--------6   Middle Sensor---------------9   Right Sensor------------11



Note: the connection should to be careful, not wrong line, after connected, Download routine to the board. Test Routine:

```
*****************************************************************************
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#define SensorLeft    6   //左感測器輸入腳
#define SensorMiddle  9   //中感測器輸入腳 #define
SensorRight   11   //右感測器輸入腳 unsigned char
SL;       //左感測器狀態 unsigned char SM;       //中
感測器狀態 unsigned char SR;       //右感測器狀態
unsigned char old_SL,old_SM,old_SR;//上一次传感器的状态 LiquidCrystal_I2C
lcd(0x27,16,2); // 定义了一个液晶
void LCD1602_init(void) {
  lcd.init();      // 初始化液晶
  lcd.backlight(); //开背光
  lcd.clear();      //清屏
}
void Sensor_IO_Config()//循迹模块 IO 初始化
{
  pinMode(SensorLeft,INPUT);   pinMode(SensorMiddle,INPUT);
  pinMode(SensorRight,INPUT);
}
```

```
void Sensor_Scan(void)//传感器状态读取函数
{
  SL = digitalRead(SensorLeft);
  SM = digitalRead(SensorMiddle);
  SR = digitalRead(SensorRight);
}
void setup()
{
   LCD1602_init(); //1602 液晶初始化
   Sensor_IO_Config();//循迹模块 IO 初始化
}

void loop()
{

  Sensor_Scan();//传感器状态读入   if(old_SL!=SL||old_SM!=SM||old_SR!=SR)//如过有一个传感器的装套发生变化，就更新显示。
  {     old_SL = SL;
old_SM = SM;
    old_SR = SR;
    lcd.setCursor(0, 0); //设置光标，从第一行，第一列开始显示
lcd.print("SL=");     lcd.print(old_SL,DEC);     lcd.print(" SM =");
lcd.print(old_SM,DEC);     lcd.print(" SR =");
lcd.print(old_SR,DEC);
  }
}
```

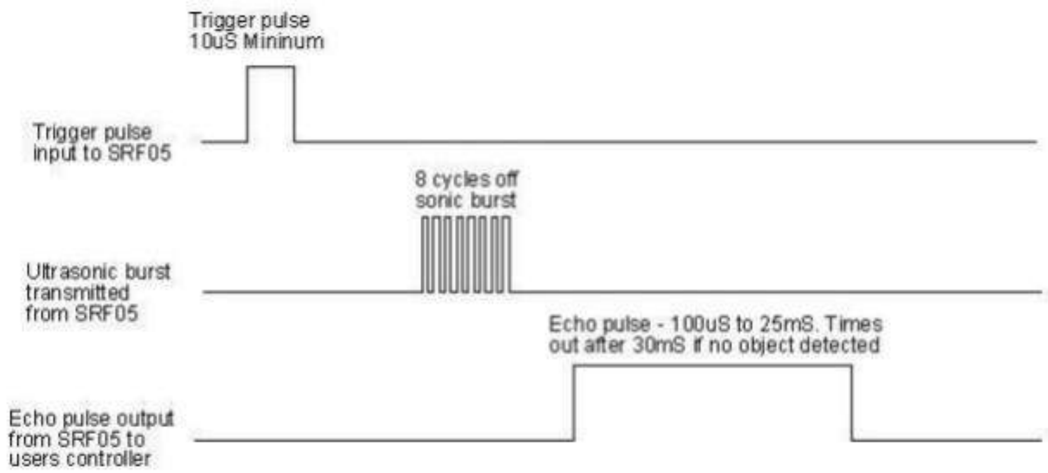
****

2. Ultrasonic Module Test



The work principle of the ultrasonic:

Let’s see how it works:



1. First we pull TRIG low, then at least 10us high level signal to trigger

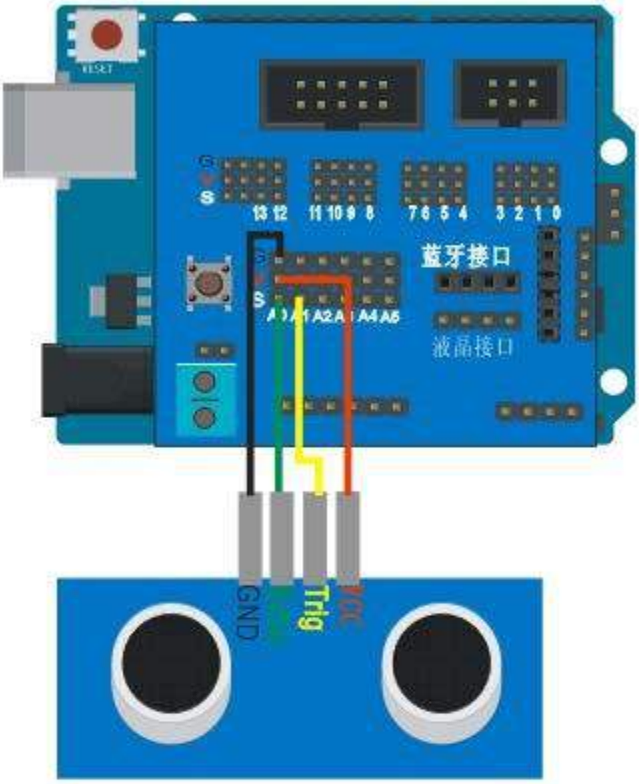
2. After triggering, module will automatically launch 8 40KHZ square waves, and automatically detect signal return

3. If the signal return, a high electrical level is output by ECHO, The duration of the high level is the time of ultrasound from the transmitter to the receiver. Then the test distance = high electrical level duration *340m/s*0.5.

4． electrical parameters

Working Voltage：0.5V(DC) Working Current：15mA
Tracing Distance：2-450cm Detection Angle：15 degree Input
the trigger pulse：10us TTL electrical level
Output echo signal: output TTL electrical level（high），Proportional to the range.

Connection Circuit diagram：



Note: before this, should first make liquid crystal test,and the circuit should be retained,because the measuring, the results are displayed in the LCD1602.

The test procedure is as follows, please connect, seriously do every step, to lay the foundation for success.

```
******************************************************************************
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
LiquidCrystal_I2C lcd(0x27,16,2);  // set the LCD address to 0x27 for a 16 chars and 2
int inputPin=A0;  // define ultrasonic signal receiver pin  ECHO to D4  int
outputPin=A1; // define ultrasonic signal transmitter pin  TRIG to D5   void setup()
 {
   Serial.begin(9600);
    lcd.init();                  // initialize the lcd       lcd.backlight();
pinMode(inputPin,   INPUT);                pinMode(outputPin,
OUTPUT);
  }
     void loop()      {
       int old_distance;        digitalWrite(outputPin,
LOW);           delayMicroseconds(2);
       digitalWrite(outputPin, HIGH); // Pulse for 10μ s to trigger ultrasonic detection
delayMicroseconds(10);         digitalWrite(outputPin, LOW);            int distance =
pulseIn(inputPin, HIGH);  // Read receiver pulse time         distance= distance/58;   //
Transform pulse time to distance          if(old_distance!=distance)
```

```
        {
          old_distance=distance;
lcd.clear();           lcd.print("H = ");
lcd.setCursor(4,                          0);
lcd.print(distance);
        }
      delay(200);
   }
```

****

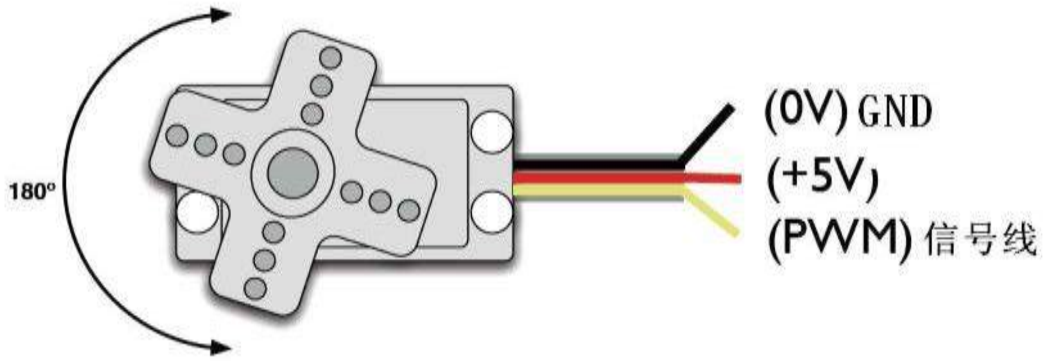
4. Servo Motor Control

The servo is a kind of position servo drive mainly composed of a casing, a circuit board, non core motor, gear and the position detector. Its working principle is to steering gear from the receiver or signal the microcontroller, the interior has a reference circuit, the cycle is 20ms, the width of the reference signal 1.5ms, compares the voltage of DC bias voltage was obtained with potentiometer, voltage difference output. Through the circuit board IC judge the direction of rotation, and then drives the non core motor starts to rotate, the power is transmitted to the swing arm through the reduction gear, at the same time by the position detector to signal, whether we've reached the positioning. Applies to those control systems need to angle changing. When the motor speed must, through the reduction gear drive cascading rotary potentiometers, the voltage difference is 0, the motor stops rotating. The rotation angle range of steering gear is 0 degrees to 180 degrees.

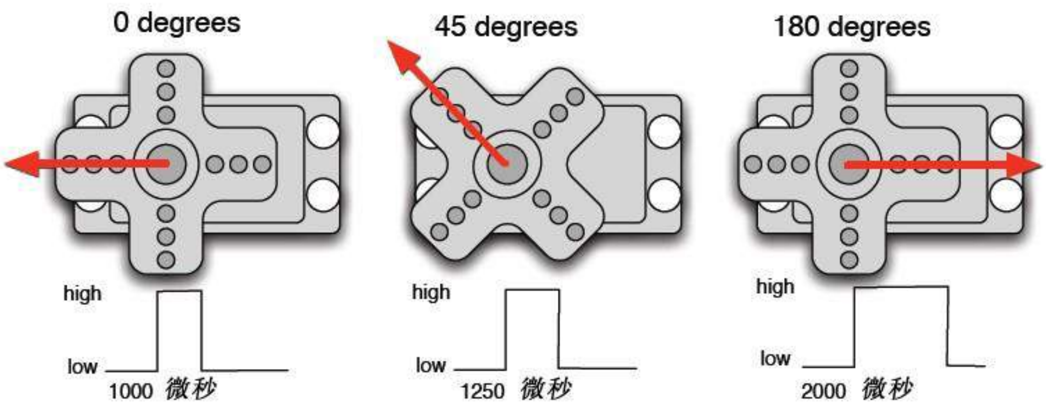


The servo has many specifications, but all of the actuator are connected respectively with three lines, brown, red,

orange three colors to distinguish, as the steering gear with different brand, color will vary, brown for the grounding line, red for the positive power supply line, orange line.

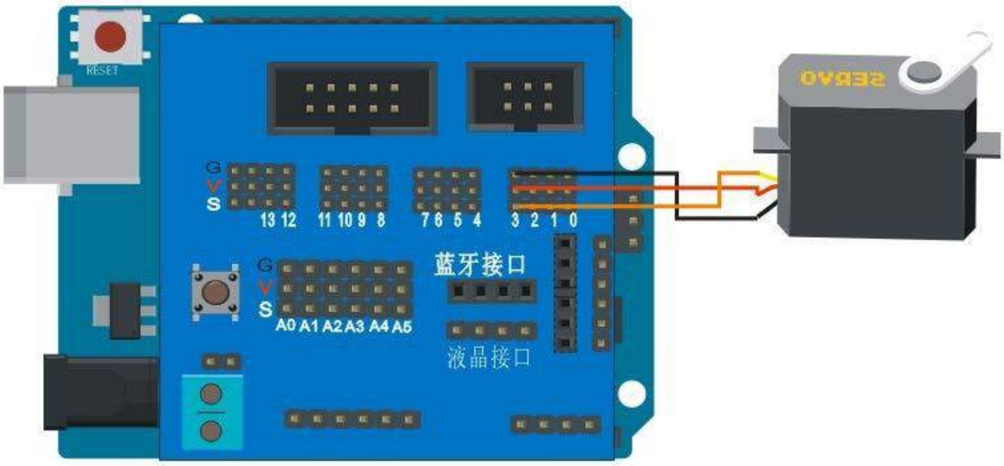


The rotation angle is regulated by PWM (pulse width modulation) signal duty cycle to achieve the standard, PWM (pulse width modulation) signal cycle is fixed at 20ms (50Hz), pulse width distribution theory should be between 1ms to 2ms, but, in fact the pulse width is from 0.5ms to 2.5ms, angular width and the 0 ° ~ 180 ° should be relatively. It is noteworthy places, because the brand is different, for the same signal, different brands of steering gear rotation angle will be different.



To understand the basic knowledge we can learn to control a servo, need the components need only a few actuator, one jumper is ok.

Connect to the Arduinuo pin 3



程序：

```
********************************************************************************
int servopin=3;//定义数字接口 3 连接伺服舵机信号线
int myangle;//定义角度变量
int pulsewidth;//定义脉宽变量
int val;
void servopulse(int servopin,int myangle)//定义一个脉冲函数
{
pulsewidth=(myangle*11)+500;//将角度转化为 500-2480 的脉宽值
digitalWrite(servopin,HIGH);//将舵机接口电平至高 delayMicroseconds(pulsewidth);//延时脉宽
值的微秒数
digitalWrite(servopin,LOW);//将舵机接口电平至低 delay(20-pulsewidth/1000);
} void
setup()
{
pinMode(servopin,OUTPUT);//设定舵机接口为输出接口
Serial.begin(9600);//连接到串行端口，波特率为 9600
Serial.println("servo=o_seral_simple ready" ) ;
}
void loop()//将 0 到 9 的数转化为 0 到 180 角度，并让 LED 闪烁相应数的次数
{
val=Serial.read();//读取串行端口的值
if(val>'0'&&val<='9')
{
val=val-'0';//将特征量转化为数值变量
val=val*(180/9);//将数字转化为角度
Serial.print("moving servo to ");
Serial.print(val,DEC); Serial.println();
for(int i=0;i<=50;i++) //给予舵机足够的时间让它转到指定角度
{
```

```
servopulse(servopin,val);//引用脉冲函数
}
}
}
```

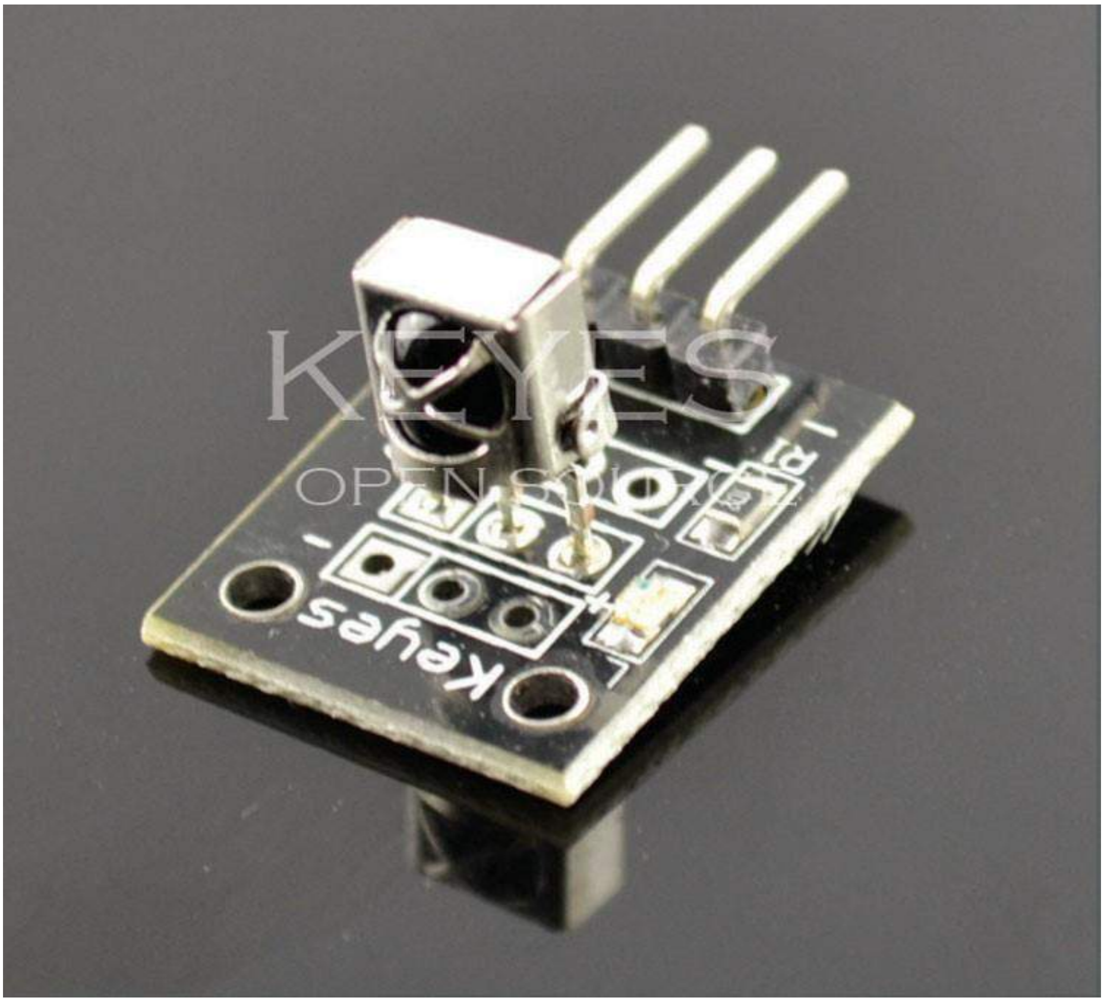
****

5. The infrared receiving module test

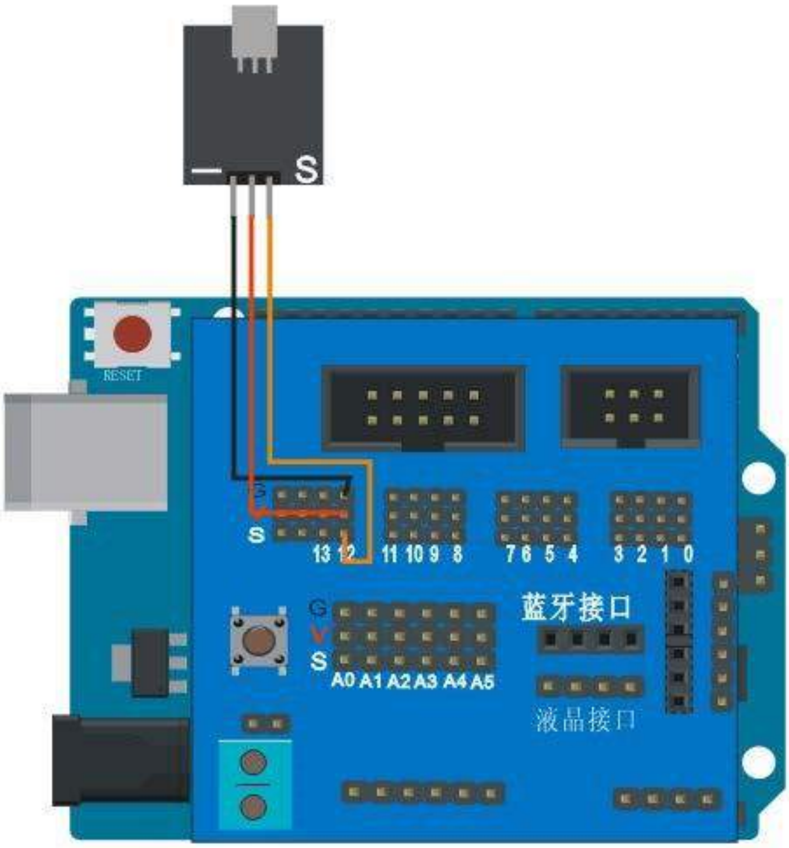
Before experiment you should know that：

1. The IRremote Library folder into the Arduino libraries directory.

2. Open the IrReceive.pde measured their infrared remote control code (in the Serial Monitor can display IRcode), then IRcode is recorded, and then to the program code can be modified into their infrared.



Infrared receiver of the connection is very simple, only one signal line is connected to the 12
Connection Circuit Diagram：



Note: this experiment to use LCD1602, so have to do LCD1602 testing, and the circuit to keep good. The test procedure is as follows.

****

```
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <IRremote.h>
LiquidCrystal_I2C lcd(0x27,16,2);
int RECV_PIN = 12;
IRrecv irrecv(RECV_PIN);
decode_results results; void
setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn(); // Start the receiver   lcd.init();
// initialize the lcd       lcd.backlight();     lcd.clear();
lcd.setCursor(0, 0);   lcd.print("code:");
} void loop()
{
  if (irrecv.decode(&results)) {

    lcd.setCursor(6, 0);     lcd.print("
");     lcd.setCursor(6, 0);
lcd.print(results.value, HEX);
    irrecv.resume(); // Receive the next value
  }
```

}

****

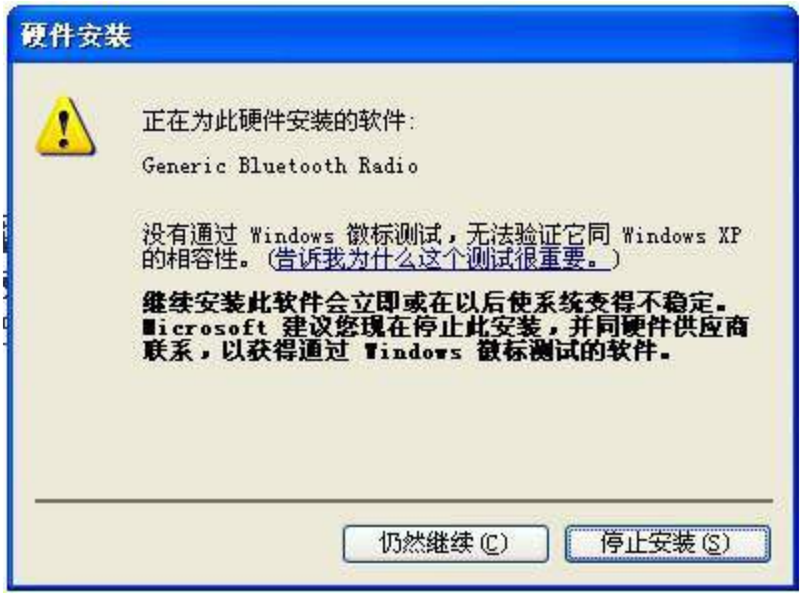
6. Bluetooth Serial Module Test



In fact, the Bluetooth serial port, have put Bluetooth serial process, install the drivers, we can regard it as a serial port to use. We talk about the PC machine driven how to install Bluetooth adapter, and connect to the Arduino.

Install the Bluetooth adapter to the USB port on the computer. Dialog box of install the driver will pop up。

Click still continue.





2. After installed,The Bluetooth icon is in thelower right corner computer, double-click it appears the following dialog.



3.upload Bluetooth Test code into the ARDUINO mian board

****

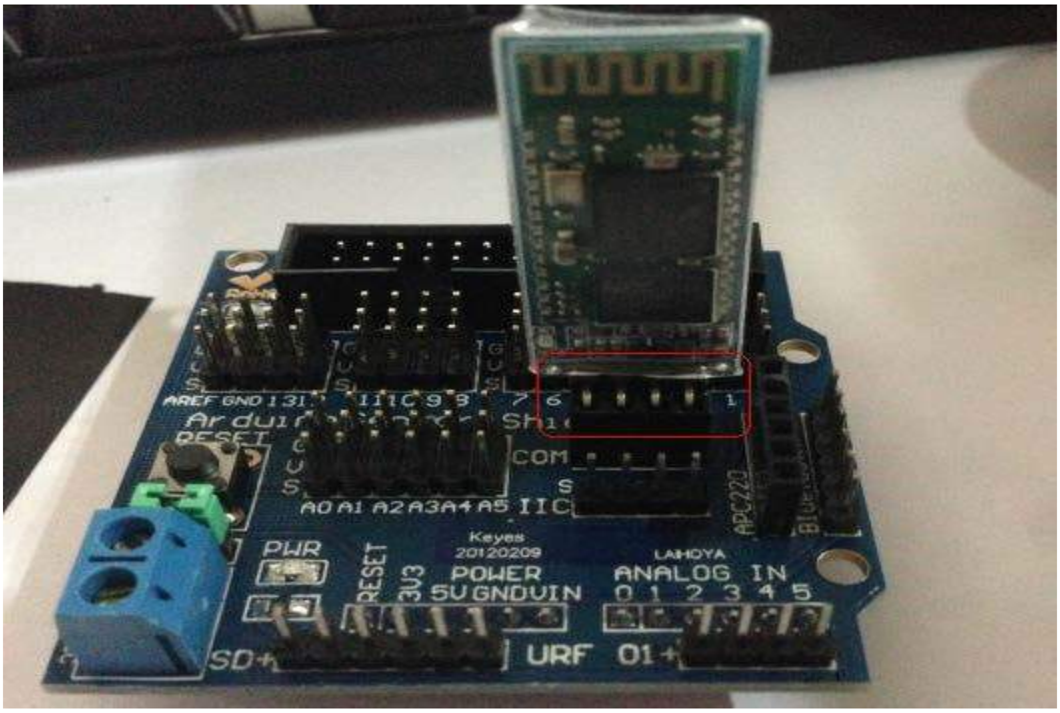
#include <LiquidCrystal_I2C.h>   //包含 I2C-LCD1602 液晶的库

```
#include <Wire.h>               //包含  I2C 的库 unsigned char val;
//定义变量  val LiquidCrystal_I2C lcd(0x27,16,2);  //定义一个
液晶
void LCD1602_init(void)           //液晶初始化用到的函数，包罗在这里
{
 lcd.init();  //调用  LiquidCrystal_I2C.h 里的  LCD 初始化函数
delay(10);   //延时  10 毫秒   lcd.backlight(); //开  LCD1602 的背
光灯   lcd.clear();     //清除屏幕
}       void
setup()
{
  LCD1602_init();
  Serial.begin(9600);   //初始化串口，其实蓝牙已经充当串口的角色。设置波特率为  9600
  lcd.setCursor(0, 0);  //光标设置在第一行，第一列，也就是左上角    lcd.print("
Bluetooth Test ");
}  void
loop()
{
  if(Serial.available()) //这一句是用来判断串口是否接收到数据的标志位。
  {
    val=Serial.read();   //读取（蓝牙）串口数据，并把值赋给  val;      lcd.setCursor(0, 1);//光标
设置在第   2 行，第一列，也就是左下角        lcd.write(val);       //把接收到的字符，显示在
LCD1602 上,注意只是单字符，不是字符串。
  }
}
```

The Bluetooth serial module is inserted into the board. And then charge to the board, and LCD1602 is connected.



The specific method refer to the I2CLCD1602 connection method. After power on, the red light will always shine.

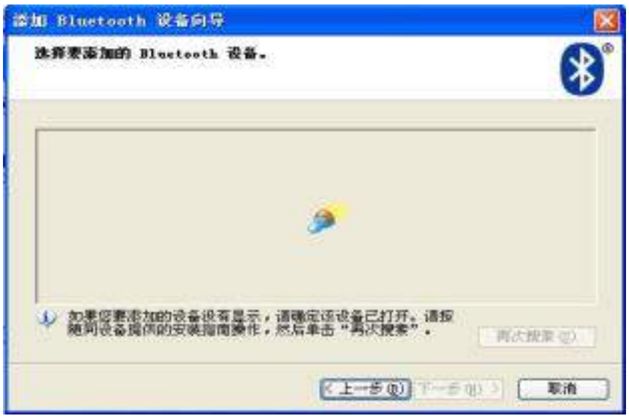


then

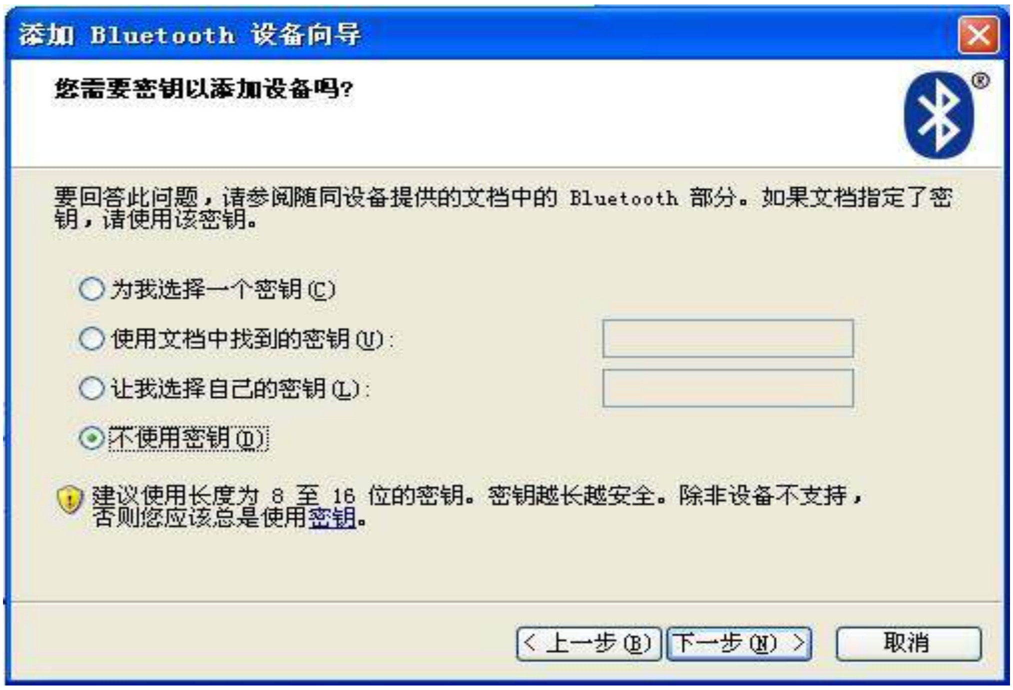




select， and then click the next botton


添加 Bluetooth 设备向导
选择要添加的 Bluetooth 设备。
如果您要添加的设备没有显示，请确定该设备已打开。请按随同设备提供的安装指南操作，然后单击“再次搜索”。
再次搜索 (S)
< 上一步 (B)
下一步 (N) >
取消


添加 Bluetooth 设备向导
选择要添加的 Bluetooth 设备。
HC-06
新设备
如果您要添加的设备没有显示，请确定该设备已打开。请按随同设备提供的安装指南操作，然后单击“再次搜索”。
再次搜索 (S)
< 上一步 (B)
下一步 (N) >
取消


添加 Bluetooth 设备向导
您需要密钥以添加设备吗?
要回答此问题，请参阅随同设备提供的文档中的 Bluetooth 部分。如果文档指定了密钥，请使用该密钥。
为我选择一个密钥 (C)
使用文档中找到的密钥 (U):
让我选择自己的密钥 (L):
不使用密钥 (D)
建议使用长度为 8 至 16 位的密钥。密钥越长越安全。除非设备不支持，否则您应该总是使用密钥。
< 上一步 (B)
下一步 (N) >
取消

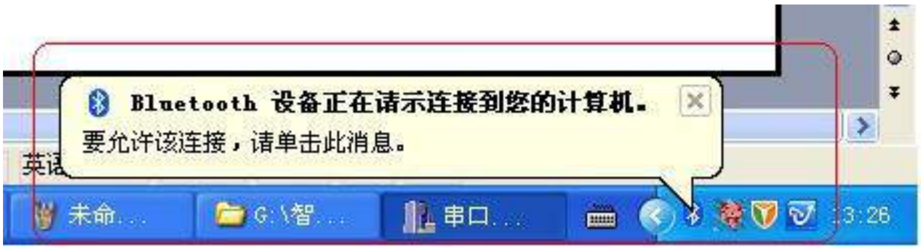

添加 Bluetooth 设备向导
Windows 正在安装设备。
正在连接...
正在安装 Bluetooth 设备...
< 上一步 (B)
下一步 (N) >
取消


添加 Bluetooth 设备向导
正在完成添加 Bluetooth 设备向导
成功地将 Bluetooth 设备连接到计算机。计算机和设备相互靠近时就可以通信。
分配给设备的 COM (串行)端口。
传出 COM 端口: COM22
传入 COM 端口: COM28
了解 Bluetooth COM 端口的更多信息。
要关闭此向导，请单击“完成”。
< 上一步 (B)
完成
取消





Double click the Bluetooth icon after the information appears.



Enter the default password: 1234





Then click on the lower right corner of the Bluetooth icon, select HC-06 and then click Properties.



select，click apply



Click confirm.



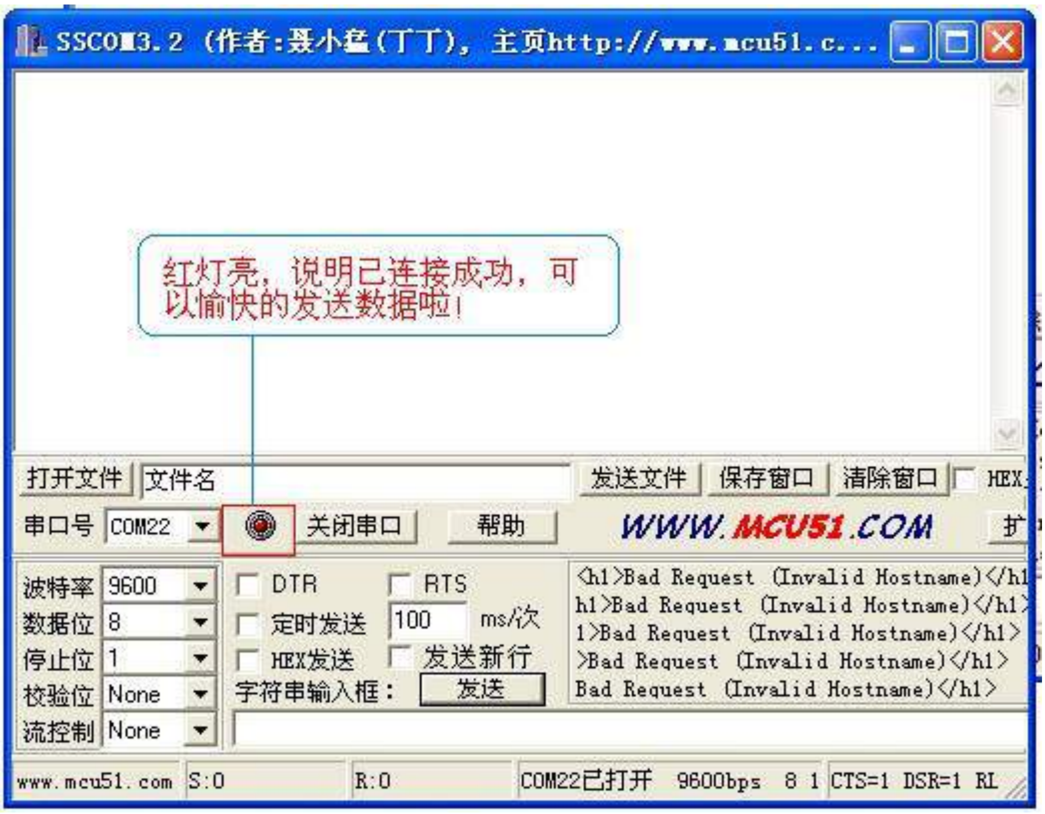
Click confirm



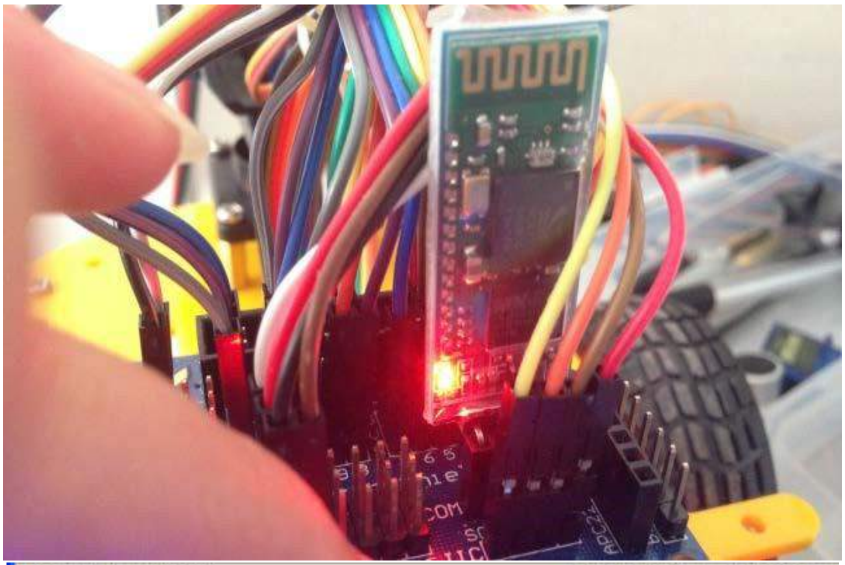
So, the drive is installed, open the new serial assistant.



Wait a moment until the connection is successful.



After a successful connection Bluetooth serial board is always bright, no flicker.





Receiving the character A, and show it, that Bluetooth has been normal work! Then can try other characters. Put the mouse on a dialog box, click, and then press any key, the key which transmits codes! Don’t need to click send every time.



7. Application of L298N motor driver board

L298N bridge driver board instructions please refer to (L298N double H bridge DC motor driver board manual), no longer saying here, but there are some users do not know how to control two DC motors, here is a detailed description.

The first VMS drive electric can be connected with an external power supply, is generally about 9V no more appropriate, the logical part of plate to take power, the terminal can be suspended, can also access +5V-+7V. Terminal three pin of two rows were used to control the DC motor. EA, EB access ArduinoPWM interface for motor speed control, I1, I2, I3, I4 interface, are used to control the two DC motor forward and backward, steering and braking, digital interface can only access Arduino.

This preparatory work basically completed, can write a program, here I write the car function of going straight, back, turning left, turning right, brake into the program process sequence for your reference.

Here is the program：

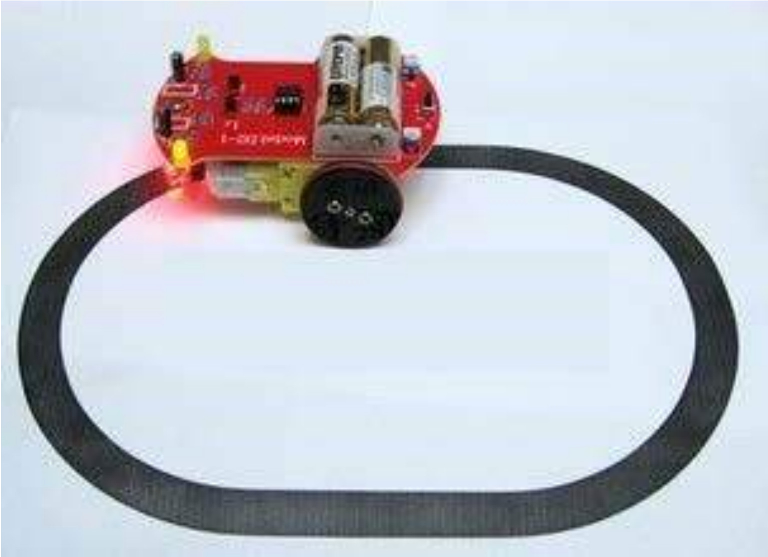
```
*******************************************************************************
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
LiquidCrystal_I2C lcd(0x27,16,2);
////////////////////////////////////////////////////////////
int pinLB=2;     // 定義  2 腳位 左後 int
pinLF=4;     // 定義  4 腳位 左前 int
pinRB=7;    // 定義  7 腳位  右後
int pinRF=8;    // 定義  8 腳位  右前
//////////調速////////////////////
int Lpwm_pin = 10;    //在這調速      //
int Rpwm_pin = 5;    //在這調速      //
///////////////////////////////
int Car_state=0;

//////////////////////////////////////////////////////////// void
advance()     // 前進
   {
    digitalWrite(pinRB,HIGH);  // 使馬達（右後）動作
digitalWrite(pinRF,LOW);      digitalWrite(pinLB,HIGH);  //
使馬達（左後）動作      digitalWrite(pinLF,LOW);
Car_state = 1;       show_state();
   }
void turnR()        //右轉(雙輪)
   {
    digitalWrite(pinRB,LOW);  //使馬達（右後）動作
```

```
    digitalWrite(pinRF,HIGH);
digitalWrite(pinLB,HIGH);      digitalWrite(pinLF,LOW);  //
使馬達（左前）動作
    Car_state = 4;      show_state();
   }
void turnL()       //左轉(雙輪)
   {
    digitalWrite(pinRB,HIGH);
    digitalWrite(pinRF,LOW);   //使馬達（右前）動作      digitalWrite(pinLB,LOW);
//使馬達（左後）動作      digitalWrite(pinLF,HIGH);      Car_state = 3;
show_state();     }     void stopp()        //停止
{
    digitalWrite(pinRB,HIGH);
digitalWrite(pinRF,HIGH);
digitalWrite(pinLB,HIGH);
digitalWrite(pinLF,HIGH);
Car_state = 5;      show_state();
}
void back()         //後退
{
    digitalWrite(pinRB,LOW);  //使馬達（右後）動作
    digitalWrite(pinRF,HIGH);
    digitalWrite(pinLB,LOW);  //使馬達（左後）動作
    digitalWrite(pinLF,HIGH);
Car_state = 2;      show_state() ;
}
void M_Control_IO_config(void)
{
  pinMode(pinLB,OUTPUT);      //      腳     位          2
pinMode(pinLF,OUTPUT);      //      腳     位          4
pinMode(pinRB,OUTPUT);      //      腳     位          7
pinMode(pinRF,OUTPUT);      //      腳     位          8
pinMode(Lpwm_pin,OUTPUT); // 腳位 10 (PWM)
  pinMode(Rpwm_pin,OUTPUT); // 腳位 5 (PWM)
} void Set_Speed(unsigned char Left,unsigned char
Right)
{
  analogWrite(Lpwm_pin,Left);
  analogWrite(Rpwm_pin,Right);
}
void show_state(void)
{   lcd.setCursor(0, 1);
lcd.print("State:");    switch(Car_state)
{      case 1:    lcd.print("  Go  ");
break;
```

```
    case 2:    lcd.print(" Back ");
break;      case 3:    lcd.print(" Left
");      break;      case 4:    lcd.print("
Right");      break;      case 5:
lcd.print(" stop ");      break;
default:      break;    }
} void
setup()
 {
M_Control_IO_config();
delay(500);       lcd.init();
lcd.backlight();
lcd.clear();
  Set_Speed(200,200);
} void loop()  {
advance() ;
delay(2000);   turnL() ;
delay(2000);   turnR() ;
delay(2000);   back() ;
delay(2000);   stopp();
delay(2000);
}
```

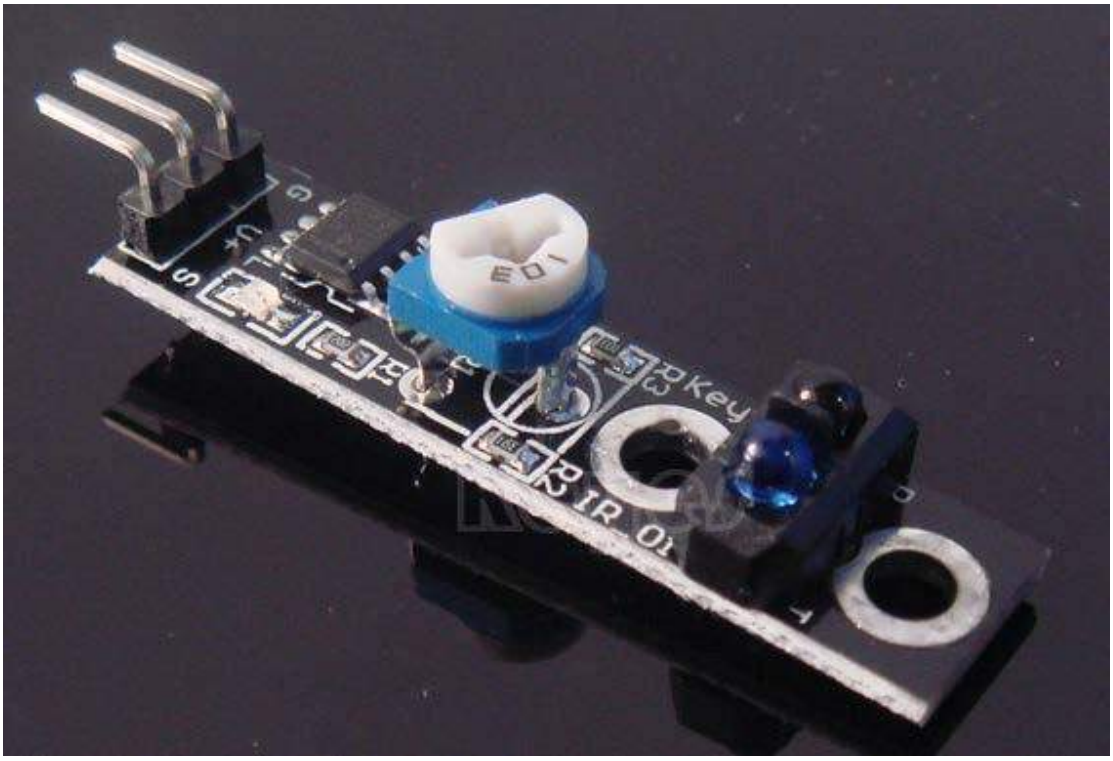
8. Tracking Smart Car



Tracking module principle: TCRT5000 infrared pipe working principle is that the infrared reflectivities of colors are not the same, the reflected signal turn into a current signal. Black and white tracing module to detect black high effective electrical level, detect white is low effective electrical level, detection height of 0 3cm.

Note: in the circuit you can use knob potentiometer to adjust the sensitivity of the black and white tracing module.

TCRT5000 infrared tubeis widely used in robot design and industrial manufacturing. TCRT5000 can be used to the production of black and white tracing robot, industrial counting sensor etc.



Method of use:

1. Sensor interface with 3 pin, respectively is GND, VCC, OUT. VCC and GND as the supply side, OUT is the signal output end.

2. Detect objects, signal output low level; not detected objects, signal output high level.

3. Mainly judge signal output end is 0 or 1. You can determine whether the object exists.

Performance parameters:

1. The detection range, detection of paper is about 2 cm. Different distance visual, color different, white the farthest.

2. The power supply voltage: 2.5V~12V, not to exceed 12V. (Note: the best use of low voltage power supply, the power supply voltage is too high the sensor lifetime becomes shorter. The 5V power supply is preferred.)

3. Working current, 5V 18~20ma. After extensive testing, the sensor hardware settings for 18~20ma working current is the best performance, mainly in the anti-interference ability.

4. To detect objects, signal output low level; not detected objects, signal output high level.

5. The sensor output TTL level, can be directly connected to the 3.3V or 5V microcontroller IO port.

Black or white line detecting principle:

1. Use the characteristics that black has small reflectivity to light, when a plane is not the color black, the sensor emits infrared light reflected back majorily. So the sensor output low level 0.

2. When a plane has a black line, sensors in the black line above the reflecting ability, because black is very weak, the reflected light is very few, not up to the action of the sensor level, so that the sensor output 1.

3. As long as we use MCU is used to judge the output end of the sensor is 0 or 1, can detect the black line.

4. The principle of detecting black as the same as white detection, white periphery color will be close to the black, and then adjust lower the sensitivity of adjustable resistance on the infrared sensor until being unable to detecte the periphery color, so that you can test line.

After understand tracking module, we can start a hunt car belongs to ourself.

This design is a kind of automatic tracing system of simple control based on Arduino, including the car system hardware and software design methods. Car takes Arduino UNO as the control core, which is used for tracing using infrared photoelectric sensor, and the pavement detection signals back to the SCM Arduino. To analyze and judge the collected signal Arduino microcontroller, timely control the drive motor to adjust the car steering, so that the car can travel along the black path automatically, realize the purpose of automatic tracing.
The design of a automatic tracing car based on DC motor, so that the car can automatically detect and track black on the ground, black car driving along a trajectory. System block diagram is as shown in figure 1-1.

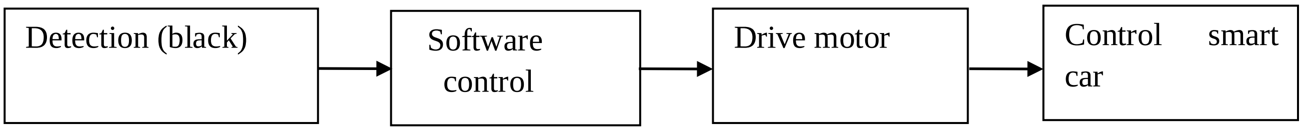


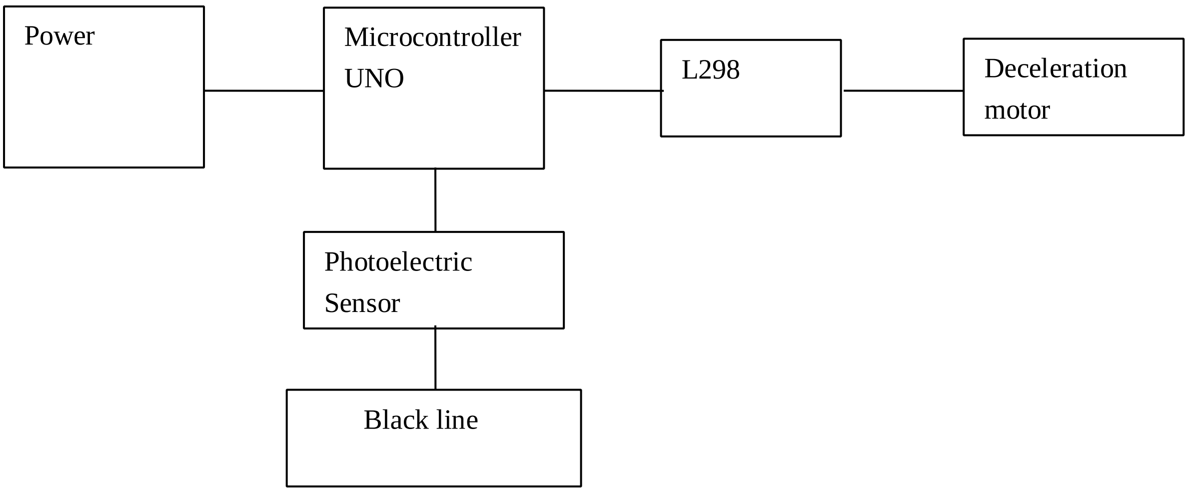
Figure 1-1 system block diagram Car tracking principle

Here the tracking refers to the car through the black line to walk in the white floor, because the black and white floor reflection coefficient of light, road can be judged according to the receiveed reflected light intensity. Method is usually taken to infrared detection method.

Infrared detection method, which uses infrared characteristics with different reflective properties on the surface of different colors, in the car driving process continually to the ground to launch the infrared, infrared light in white paper floor diffuse reflection, the reflected light is installed in the car of the receiving tube receiving; if black is infrared absorption, cars receiving tube receiving less than the infrared. SCM will receive the reflected infrared light as the basis to determine the route line position and car. Infrared detector distance is limited.

Control system design

Automatic vehicle tracking control system consists of main control module, power supply circuit, infrared detection module, motor drive module, the structure block diagram of control system is shown in figure 2-1.



Car tracking flow chart

The car entered the tracking mode, namely began constantly scanning and detector connected to the I/O port of the SCM, once detected a signal of a I/O port, enter judgment processing procedures, to determine which one of 3 detectors detect the black line.

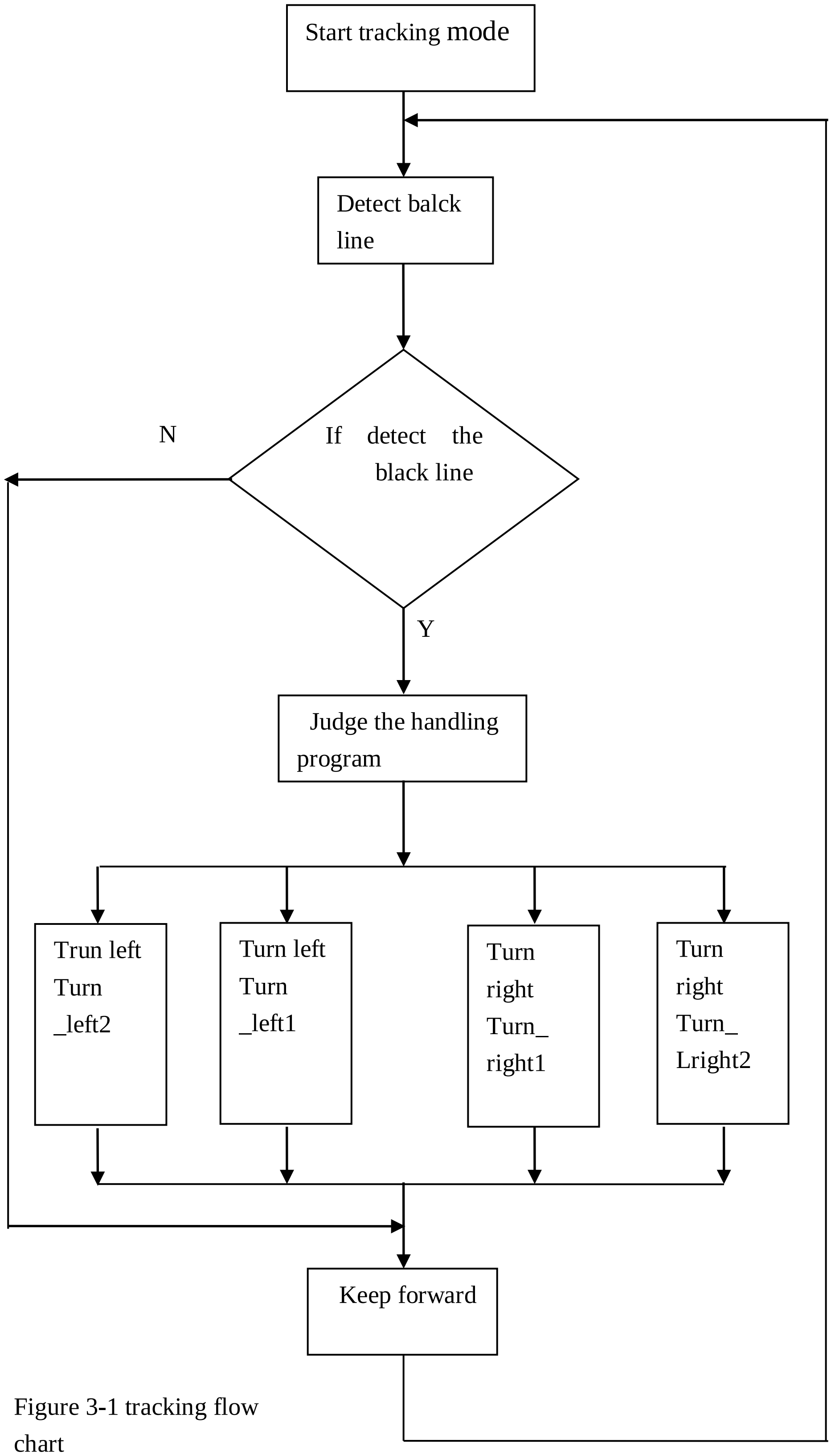


Figure 3-1 tracking flow chart

Arduino car tracking wiring diagram Vehicle

tracking Arduino program:

```
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#define SensorLeft    6   //左感測器輸入腳
#define SensorMiddle  9   //中感測器輸入腳 #define
SensorRight   11   //右感測器輸入腳 unsigned char
SL;        //左感測器狀態 unsigned char SM;        //中
感測器狀態 unsigned char SR;        //右感測器狀態
#define Lpwm_pin  5     //车速控制引脚----电机驱动板的  ENA
#define Rpwm_pin  10    //车速控制引脚----电机驱动板的  ENB
int pinLB=2;           //车轮转向控制----电机驱动板的  IN1 int
pinLF=4;           //车轮转向控制----电机驱动板的  IN2 int
pinRB=7;           //车轮转向控制----电机驱动板的  IN3 int
pinRF=8;           //车轮转向控制----电机驱动板的  IN4 unsigned
char Lpwm_val = 150;//初始化左轮速度为  150 unsigned char
Rpwm_val = 150;//初始化右轮速度为  150
int Car_state=0;            //车子运行状态
LiquidCrystal_I2C lcd(0x27,16,2);  // set the LCD address to 0x27 for a 16 chars and 2  void
LCD1602_init(void)
{
  lcd.init();
  lcd.backlight();    lcd.clear();
}
void Sensor_IO_Config()
{
  pinMode(SensorLeft,INPUT);   pinMode(SensorMiddle,INPUT);
  pinMode(SensorRight,INPUT);
}
void Sensor_Scan(void)
{
  SL = digitalRead(SensorLeft);   SM
= digitalRead(SensorMiddle);
  SR = digitalRead(SensorRight);
}
void M_Control_IO_config(void)//电机驱动板的  IO 初始化函数
{
  pinMode(pinLB,OUTPUT); // 脚位 2--电机驱动板的  IN1   pinMode(pinLF,OUTPUT);
// 脚位 4--电机驱动板的  IN2   pinMode(pinRB,OUTPUT); // 脚位 7--电机驱动板的
IN3   pinMode(pinRF,OUTPUT); // 脚位 8--电机驱动板的  IN4
pinMode(Lpwm_pin,OUTPUT); // 脚位 5  (PWM) --电机驱动板的  ENA
  pinMode(Rpwm_pin,OUTPUT); // 脚位 10 (PWM) --电机驱动板的  ENB
}
```

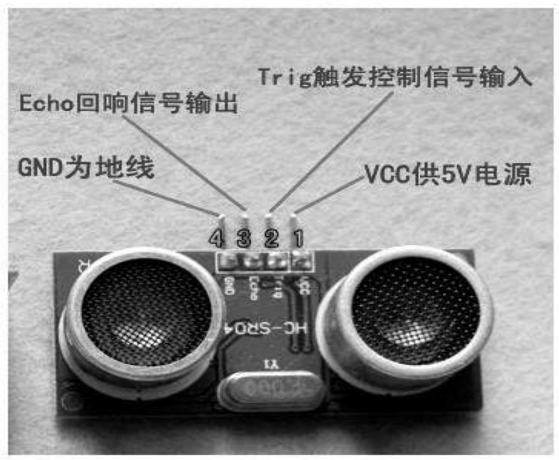
```
void Set_Speed(unsigned char Left,unsigned char Right)//车速设定函数
{
  analogWrite(Lpwm_pin,Left);
  analogWrite(Rpwm_pin,Right);
}
void advance(void)     // 前进
{
   digitalWrite(pinRB,LOW);  // 右轮前进
digitalWrite(pinRF,HIGH);
digitalWrite(pinLB,LOW);  // 左轮前进
digitalWrite(pinLF,HIGH);     Car_state = 1;
show_state();
}
void turnR(void)        //右转
{
   digitalWrite(pinRB,LOW);  //左轮前进
digitalWrite(pinRF,HIGH);    digitalWrite(pinLB,HIGH);
   digitalWrite(pinLF,LOW);  //右轮停止
   Car_state = 4;    show_state();
}
void turnL(void)        //左转
{
   digitalWrite(pinRB,HIGH);       // 左 轮 停 止
digitalWrite(pinRF,LOW);
digitalWrite(pinLB,LOW);    //右轮前进
   digitalWrite(pinLF,HIGH);
Car_state = 3;    show_state(); }     void
stopp(void)             //停止
{
   digitalWrite(pinRB,HIGH);  //左右轮都停止
   digitalWrite(pinRF,HIGH);
digitalWrite(pinLB,HIGH);
digitalWrite(pinLF,HIGH);
Car_state = 5;    show_state();
}
void back(void)           //後退
{
   digitalWrite(pinRB,HIGH);  //左右轮都退后
   digitalWrite(pinRF,LOW);
digitalWrite(pinLB,HIGH);
digitalWrite(pinLF,LOW);
Car_state = 2;    show_state() ;
}
void show_state(void)  //显示小车当时状态
{
```

```
 lcd.setCursor(0, 1); //从第二行开始显示
switch(Car_state)   {     case 1:lcd.print(" Go
");Serial.print("\n GO");     break;     case
2:lcd.print("Back ");Serial.print("\n Back");     break;
   case 3:lcd.print("Left ");Serial.print("\n Left");
break;     case 4:lcd.print("Right");Serial.print("\n
Right");     break;
   case 5:lcd.print("Stop ");Serial.print("\n Stop");
   break;     default:
break;
  }
} void
setup()  {
  LCD1602_init();
  Sensor_IO_Config();
  M_Control_IO_config();       //电机控制模块 IO 初始化   Set_Speed(Lpwm_val,Rpwm_val);
//车速初始设定
  lcd.clear();
  lcd.setCursor(0, 0);  //光标设置在第一行，第一列，
lcd.print("  Wait  Signal  ");   stopp(); }
unsigned char old_SL,old_SM,old_SR;
void loop()  {
 Sensor_Scan();
if((SL==1&&SM==1&&SR==1)||(SL==1&&SM==1&&SR==0)||(SL==0&&SM==1&&SR==1)
)advance();
 if(SL==1&&SM==0&&SR==0)turnL();   if(SL==0&&SM==0&&SR==1)turnR();
 if(SL==0&&SM==0&&SR==0)
 {
   if(old_SL==1&&old_SM==0&&old_SR==0)
   {
     while(!(SL==1&&SM==1&&SR==0))
     {
     Sensor_Scan();
turnL();      }     }
   else if(old_SL==0&&old_SM==0&&old_SR==1)
   {
    while(!(SL==0&&SM==1&&SR==1))
     {
       Sensor_Scan();        turnR();
     }
   }   }
 if(old_SL!=SL||old_SM!=SM||old_SR!=SR)
 {     old_SL = SL;
old_SM = SM;
   old_SR = SR;                        ;   }
}
```

8. Ultrasonic Intelligent Vehicle

Obstacle avoidance is easy to implement, simple calculation, easy to achieve real-time control of intelligent ultrasonic, and can meet the practical requirements in terms of measurement accuracy, thus become the obstacle avoidance methods commonly used. Ultrasonic using method refers to (Arduino ultrasonic ranging description).

Ultrasonic intelligent wiring diagram




```
#include <LiquidCrystal_I2C.h>   //包含  I2C-LCD1602 液晶的库
#include <Wire.h>                //包含  I2C 的库 int
inputPin=A0;  // receiver pin  ECHO to A0 int
outputPin=A1; // transmitter pin  TRIG to A1
unsigned char Bluetooth_val;       //定义变量  val
LiquidCrystal_I2C lcd(0x27,16,2);  //定义一个液晶
```

```
#define Lpwm_pin  5   //在這調速  #define
Rpwm_pin  10   //在這調速 // int pinLB=2;
// 定義  2 腳位 左後 int pinLF=4;     // 定
義  4 腳位 左前 int pinRB=7;    // 定義  7
腳位  右後
int pinRF=8;     // 定義   8 腳位   右前
unsigned    char    Lpwm_val    =    150;
unsigned   char   Rpwm_val   =   150;   int
Car_state=0;
int servopin=3;//定义数字接口  9 连接伺服舵机信号线
int myangle;//定义角度变量 int
pulsewidth;//定义脉宽变量
unsigned char DuoJiao=90;//
void servopulse(int servopin,int myangle)//定义一个脉冲函数
{
pulsewidth=(myangle*11)+500;//将角度转化为  500-2480 的脉宽值
digitalWrite(servopin,HIGH);//将舵机接口电平至高 delayMicroseconds(pulsewidth);//延时脉宽
值的微秒数
digitalWrite(servopin,LOW);//将舵机接口电平至低
delay(20-pulsewidth/1000);
}

void Set_servopulse(int set_val)
{
 for(int i=0;i<=10;i++) //给予舵机足够的时间让它转到指定角度
   servopulse(servopin,set_val);//引用脉冲函数
} void
M_Control_IO_config(void)
{
  pinMode(pinLB,OUTPUT);      //      腳     位          2
pinMode(pinLF,OUTPUT);      //      腳     位          4
pinMode(pinRB,OUTPUT);      //      腳     位          7
pinMode(pinRF,OUTPUT);      //      腳     位          8
pinMode(Lpwm_pin,OUTPUT); // 腳位 11 (PWM)
  pinMode(Rpwm_pin,OUTPUT); // 腳位 10 (PWM)
}
void Set_Speed(unsigned char Left,unsigned char Right)
{
  analogWrite(Lpwm_pin,Left);
  analogWrite(Rpwm_pin,Right);
}
void advance()     // 前進
    {
```

```
    digitalWrite(pinRB,LOW);  // 使馬達（右後）動作      digitalWrite(pinRF,HIGH);
    digitalWrite(pinLB,LOW);  // 使馬達（左後）動作
digitalWrite(pinLF,HIGH);       Car_state = 1;
show_state();       }
void turnR()       //右轉(雙輪)
   {
    digitalWrite(pinRB,LOW);  //使馬達（右後）動作
digitalWrite(pinRF,HIGH);      digitalWrite(pinLB,HIGH);
    digitalWrite(pinLF,LOW);  //使馬達（左前）動作
    Car_state = 4;
    show_state();     }
void turnL()       //左轉(雙輪)
   {
    digitalWrite(pinRB,HIGH);      digitalWrite(pinRF,LOW
);   //使馬達（右前）動作
    digitalWrite(pinLB,LOW);   //使馬達（左後）動作
    digitalWrite(pinLF,HIGH);
Car_state  =  3;        show_state();
}     void stopp()        //停止
   {
digitalWrite(pinRB,HIGH);
digitalWrite(pinRF,HIGH);
digitalWrite(pinLB,HIGH);
digitalWrite(pinLF,HIGH);
Car_state = 5;      show_state();
}
void back()          //後退
   {
    digitalWrite(pinRB,HIGH);  //使馬達（右後）動作
digitalWrite(pinRF,LOW);      digitalWrite(pinLB,HIGH);
//使馬達（左後）動作      digitalWrite(pinLF,LOW);
Car_state = 2;      show_state() ;         } void
show_state(void)
{   lcd.setCursor(0,
1);
switch(Car_state)    {
    case 1:lcd.print(" Go  ");Serial.print("\n GO");
break;      case 2:lcd.print("Back ");Serial.print("\n
Back");      break;
    case 3:lcd.print("Left ");Serial.print("\n Left");
break;      case 4:lcd.print("Right");Serial.print("\n
Right");      break;
    case 5:lcd.print("Stop ");Serial.print("\n Stop");       break;
    default:
break;    }
}
```

```
void LCD1602_init(void)          //液晶初始化用到的函数，包罗在这里
{
 lcd.init(); //调用 LiquidCrystal_I2C.h 里的 LCD 初始化函数
delay(10);  //延时 10 毫秒  lcd.backlight(); //开 LCD1602 的背
光灯  lcd.clear();   //清除屏幕
}
void Show_V(unsigned char V)
{             lcd.setCursor(11,  0);
lcd.print("V=                  ");
lcd.setCursor(13,               0);
lcd.print(V,DEC);
Serial.print("\n Speed = ");
   Serial.print(V,DEC);
}
void Show_DuoJiao(unsigned char Jiao)
{                 lcd.setCursor(6,1);
lcd.print("C=                  ");
lcd.setCursor(8,                1);
lcd.print(Jiao,DEC);
Serial.print("\n JiaoDu = ");
   Serial.print(Jiao,DEC);
} void
Bluetooth_Control()
{
  lcd.setCursor(0, 0); //光标设置在第一行，第一列，也就是左上角
lcd.print("Blue_ctr      ");  Show_V(Lpwm_val);
 while(Bluetooth_val!= 'E')
 {
  if(Serial.available()) //这一句是用来判断串口是否接收到数据的标志位。
   {
    Bluetooth_val=Serial.read(); //读取（蓝牙）串口数据，并把值赋给 val;
switch(Bluetooth_val)
    {
     case '7': if(DuoJiao<=180){ //duoji  turn left
              DuoJiao+=10;
              Set_servopulse(DuoJiao);
              Show_DuoJiao(DuoJiao);}              break;
case '9': if(DuoJiao-10>=0){ //duoji turn right
              DuoJiao-=10;
              Set_servopulse(DuoJiao);              Show_DuoJiao(DuoJiao);}
     break;          case '8':advance();
//UP       break;       case '2': back();
//back      break;      case '4':turnL();
//Left      break;      case '6':turnR();
```

```
//Right      break;      case '5':stopp();
//stop
      break;       case
'3':if(Rpwm_val+10<=250&&Rpwm_val+10<=250){
                Lpwm_val+=10; Rpwm_val+=10;
                Set_Speed(Lpwm_val,Rpwm_val);
                Show_V(Lpwm_val);
                }       break;
      case '1':if(Rpwm_val-10>=0&&Rpwm_val-10>=0){
Lpwm_val-=10; Rpwm_val-=10;
                Set_Speed(Lpwm_val,Rpwm_val);
                Show_V(Lpwm_val);
                }       break;       case '0':
Ultrasonic_Ranging(1);
      break;       default:stopp();
break;
     }
    }   }
lcd.clear();
  lcd.setCursor(0, 0);  //光标设置在第一行，第一列，
lcd.print("  Wait  Signal  ");   stopp();
}

void Self_Control(void)
{
  int H;
  lcd.setCursor(0, 0);  //光标设置在第一行，第一列，也就是左上角
lcd.print("Self_Ctr        ");   Show_V(Lpwm_val);
  while(Bluetooth_val!= 'E')
  {
    if(Serial.available()) //这一句是用来判断串口是否接收到数据的标志位。
Bluetooth_val=Serial.read();  //读取（蓝牙）串口数据，并把值赋给  val;
    H = Ultrasonic_Ranging(1);     if(H<25)
{        stopp();
while(Ultrasonic_Ranging(1)<50)
       {
         if(Ultrasonic_Ranging(1)<15)
         {            back();
delay(500);           }
else            {
turnL();
delay(1000);           }
       }
    } advance();
  }   lcd.clear();
```

```
  lcd.setCursor(0, 0);  //光标设置在第一行，第一列，     lcd.print("
Wait  Signal  ");
  stopp();
}
int Ultrasonic_Ranging(unsigned char Mode)
{    int old_distance;
digitalWrite(outputPin, LOW);
delayMicroseconds(2);
  digitalWrite(outputPin, HIGH); // Pulse for 10μ s to trigger ultrasonic detection
delayMicroseconds(10);    digitalWrite(outputPin, LOW);       int distance =
pulseIn(inputPin, HIGH);// Read receiver pulse time    distance= distance/58;   //
Transform pulse time to distance       if(Mode==1){          lcd.setCursor(11, 1);
        lcd.print("H=    ");          lcd.setCursor(13,
1);          lcd.print(distance,DEC);
        Serial.print("\n       H       =       ");
Serial.print(distance,DEC);        return
distance;
  }
  else  return distance;
}
void setup()
{
  pinMode(servopin,OUTPUT);//设定舵机接口为输出接口
  LCD1602_init();
  M_Control_IO_config();
  Set_Speed(Lpwm_val,Rpwm_val);
Set_servopulse(DuoJiao);    pinMode(inputPin,
INPUT);     pinMode(outputPin, OUTPUT);
  Serial.begin(9600);  //初始化串口，其实蓝牙已经充当串口的角色。设置波特率为 9600
  lcd.setCursor(0, 0);  //光标设置在第一行，第一列，也就是左上角
lcd.print("  Wait  Signal  ");    stopp();
}
void loop()
{
  if(Serial.available()) //这一句是用 8 来判断串口是否接收到数据的标志位。
  {
     Bluetooth_val=Serial.read();       if(Bluetooth_val=='W')
Bluetooth_Control();       if(Bluetooth_val=='S')
Self_Control();
   }     delay(20);
}
```

10. Infrared Remote Control of Intelligent Vehicle Experiment

The car wiring diagram:

The detected infrared key code, replace the infrared control part of the program

Infrared remote control of intelligent vehicle program

```
//******红外遥控智能车程序*******
#include <LiquidCrystal_I2C.h>  //包含 I2C-LCD1602 液晶的库
#include <Wire.h>              //包含 I2C 的库
#include <IRremote.h> int
RECV_PIN = 12; IRrecv
irrecv(RECV_PIN);
decode_results results;
#define IR_Go      0x00ff629d
#define IR_Back    0x00ffa857
#define IR_Left    0x00ff22dd
#define IR_Right   0x00ffc23d #define
IR_Stop    0x00ff02fd
#define IR_ESC     0x00ff52ad
LiquidCrystal_I2C lcd(0x27,16,2);  //定义一个液晶
#define  Lpwm_pin   5      //在這調速   #define
Rpwm_pin  10   //在這調速 // int pinLB=2;       //
定義  2 腳位 左後 int pinLF=4;          // 定義  4
腳位 左前 int pinRB=7;         // 定義  7 腳位  右
後
int pinRF=8;         // 定義  8 腳位  右前 unsigned
char Lpwm_val = 150; unsigned char Rpwm_val
= 150; int Car_state=0;
void M_Control_IO_config(void)
{
  pinMode(pinLB,OUTPUT);     //     腳    位         2
pinMode(pinLF,OUTPUT);     //     腳    位         4
pinMode(pinRB,OUTPUT);     //     腳    位         7
pinMode(pinRF,OUTPUT);     //     腳    位         8
pinMode(Lpwm_pin,OUTPUT); // 腳位 11 (PWM)
  pinMode(Rpwm_pin,OUTPUT); // 腳位 10 (PWM)
} void Set_Speed(unsigned char Left,unsigned char
Right)
```

```
{
  analogWrite(Lpwm_pin,Left);
analogWrite(Rpwm_pin,Right); }
void advance(void)     // 前進
{
  digitalWrite(pinRB,LOW);  // 使馬達（右後）動作
  digitalWrite(pinRF,HIGH);
  digitalWrite(pinLB,LOW);  // 使馬達（左後）動作
digitalWrite(pinLF,HIGH);     Car_state = 1;
show_state();
}
void turnR(void)        //右轉(雙輪)
{
  digitalWrite(pinRB,LOW);  //使馬達（右後）動作
  digitalWrite(pinRF,HIGH);
digitalWrite(pinLB,HIGH);
  digitalWrite(pinLF,HIGH);  //使馬達（左前）動作
  Car_state = 4;    show_state();
}
void turnL(void)        //左轉(雙輪)
{
  digitalWrite(pinRB,HIGH);
digitalWrite(pinRF,HIGH);   //使馬達（右前）動作
  digitalWrite(pinLB,LOW);   //使馬達（左後）動作
  digitalWrite(pinLF,HIGH);
Car_state = 3;    show_state();
}
void stopp(void)         //停止
{
  digitalWrite(pinRB,HIGH);
digitalWrite(pinRF,HIGH);
digitalWrite(pinLB,HIGH);
digitalWrite(pinLF,HIGH);
Car_state = 5;    show_state();
}
void back(void)          //後退
{
  digitalWrite(pinRB,HIGH);  //使馬達（右後）動作
digitalWrite(pinRF,LOW);    digitalWrite(pinLB,HIGH);
//使馬達（左後）動作    digitalWrite(pinLF,LOW);
  Car_state = 2;
  show_state() ;
}
void show_state(void)
```

```
{   lcd.setCursor(0,
1);
switch(Car_state)
  {      case 1:lcd.print(" Go  ");Serial.print("\n GO");
break;      case 2:lcd.print("Back ");Serial.print("\n
Back");      break;
    case 3:lcd.print("Left ");Serial.print("\n Left");      break;
    case 4:lcd.print("Right");Serial.print("\n Right");      break;
    case 5:lcd.print("Stop ");Serial.print("\n Stop");
break;      default:      break;    }
}
void LCD1602_init(void)           //液晶初始化用到的函数，包罗在这里
{
 lcd.init();  //调用  LiquidCrystal_I2C.h 里的  LCD 初始化函数
delay(10);   //延时  10 毫秒   lcd.backlight(); //开  LCD1602 的背
光灯
 lcd.clear();    //清除屏幕
}
void IR_Control(void)
{
  unsigned long Key;
  lcd.setCursor(0,0);  //光标设置在第一行，第一列，也就是左上角
lcd.print("IR_Ctr          ");   while(Key!=IR_ESC)
 {
  if(irrecv.decode(&results)) //这一句是用来判断串口是否接收到数据的标志位。
   {
    Key = results.value;
switch(Key)      {
      case IR_Go:advance();   //UP        break;
      case IR_Back: back();   //back        break;
      case IR_Left:turnL();   //Left        break;
      case IR_Right:turnR(); //Right        break;
      case IR_Stop:stopp();   //stop
break;        default:        break;
    }
    irrecv.resume(); // Receive the next value
   }    }
lcd.clear();
 lcd.setCursor(0, 0);  //光标设置在第一行，第一列，
 lcd.print("  Wait  Signal  ");
stopp(); }
void setup()
{
  LCD1602_init();
  M_Control_IO_config();
```

```
  Set_Speed(Lpwm_val,Rpwm_val);
  irrecv.enableIRIn(); // Start the receiver
  Serial.begin(9600);  //初始化串口，其实蓝牙已经充当串口的角色。设置波特率为 9600
  lcd.setCursor(0, 0); //光标设置在第一行，第一列，也就是左上角
lcd.print("  Wait  Signal  ");   stopp();
}
void loop()  {
  if (irrecv.decode(&results)) {
if(results.value == IR_Stop )IR_Control();
irrecv.resume(); // Receive the next value
  }
}
```

11. Mobile Phone Bluetooth Intelligent Vehicle Control

Arduino communication via Bluetooth

The name comes from the Tenth Century Danish King Harald Blatand, Blatand in English in meaning can be interpreted as Bluetooth (Bluetooth).

The Bluetooth (Bluetooth) technology, is actually a kind of short distance wireless technology, using "Bluetooth" technology, can effectively simplify the palm communication between computer, notebook computer and mobile phone and other mobile communication terminal equipment, but also can successfully simplify these devices and the Internet (Internet) between communication, so that data transmission between these modern communication equipment and Internet become more quickly and efficiently, widen the road for wireless communication.

As this is the first deal with the Bluetooth module, today was the first to small, let the success of communication between Arduino and PC. The first wiring, motherboard +5V connect to Bluetooth VCC, Bluetooth -GND motherboard motherboard GND connection, TX connection of Bluetooth RX, Bluetooth TX RX. When the Bluetooth module and the success of PC machine connected to the power supply, the power indicator exhibit of lanterns flicker Bluetooth module, connect the green light will light up.

Here to have a look for the program, I will let Arduino receive my input "R", and the pin13 interface of the LED flash, then output “hello”words.

The procedure is as follows:

```
char val; int
ledpin=13;
void setup()
{
  Serial.begin(9600);
  pinMode(ledpin,OUTPUT);
} void
loop() {
  val=Serial.read();
if(val=='r')   {
    digitalWrite(ledpin，HIGH);
delay((500);     digitalWrite(ledpin，
LOW);     delay(500);
   Serial.println("hello");

  }  }
```

Next we learn about Arduino Bluetooth remote programmable intelligent vehicle. Through the Bluetooth control forward and backward, turn left, turn right, simple button, computer and mobile phone two control modes. (Mobile phone operating system support Android version 2.3.7 and above, the computer must be built-in Bluetooth)

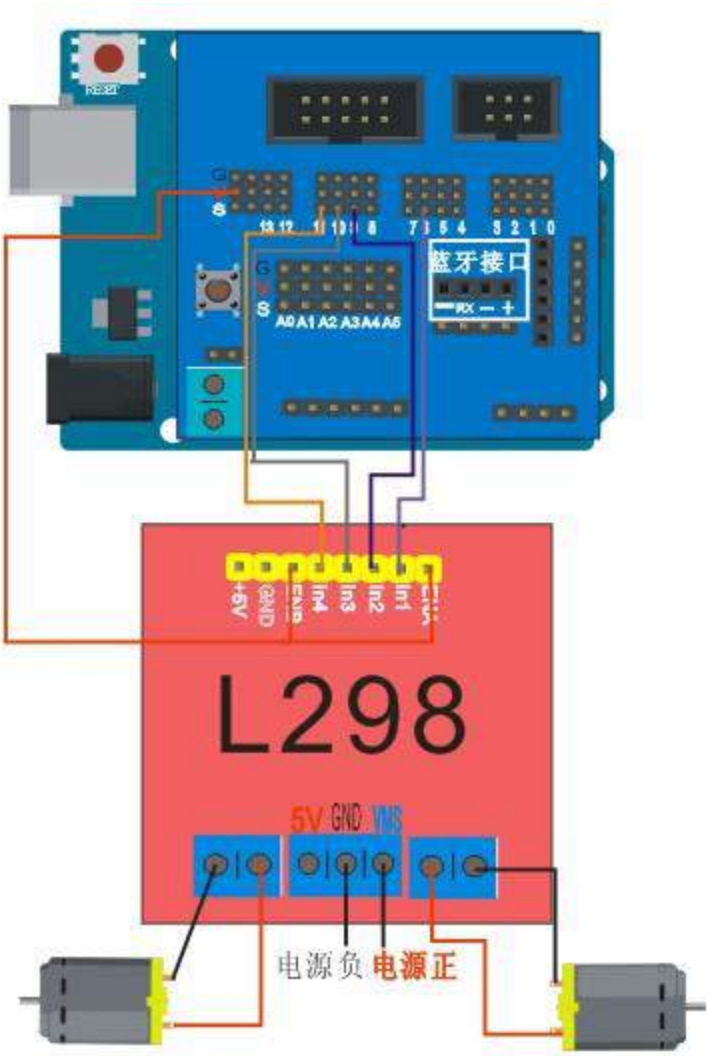
When the first use the mobile phone need to match with Bluetooth (after the first match, will not be in the location of a wireless device later), have a look at first steps as follows:

1. Remember to open the Bluetooth mobile phone Oh, open the software will remind the user to open the Bluetooth.

2. Then, as shown in the text, connect to Bluetooth devices, first scan paired bluetooth devices, otherwise cannot connect car.

3. Match with the car, password is “1234”, have a try. This is the way you can come in.

Wiring diagram;



Arduino 蓝牙遥控可编程智能车程序：

//*********************************

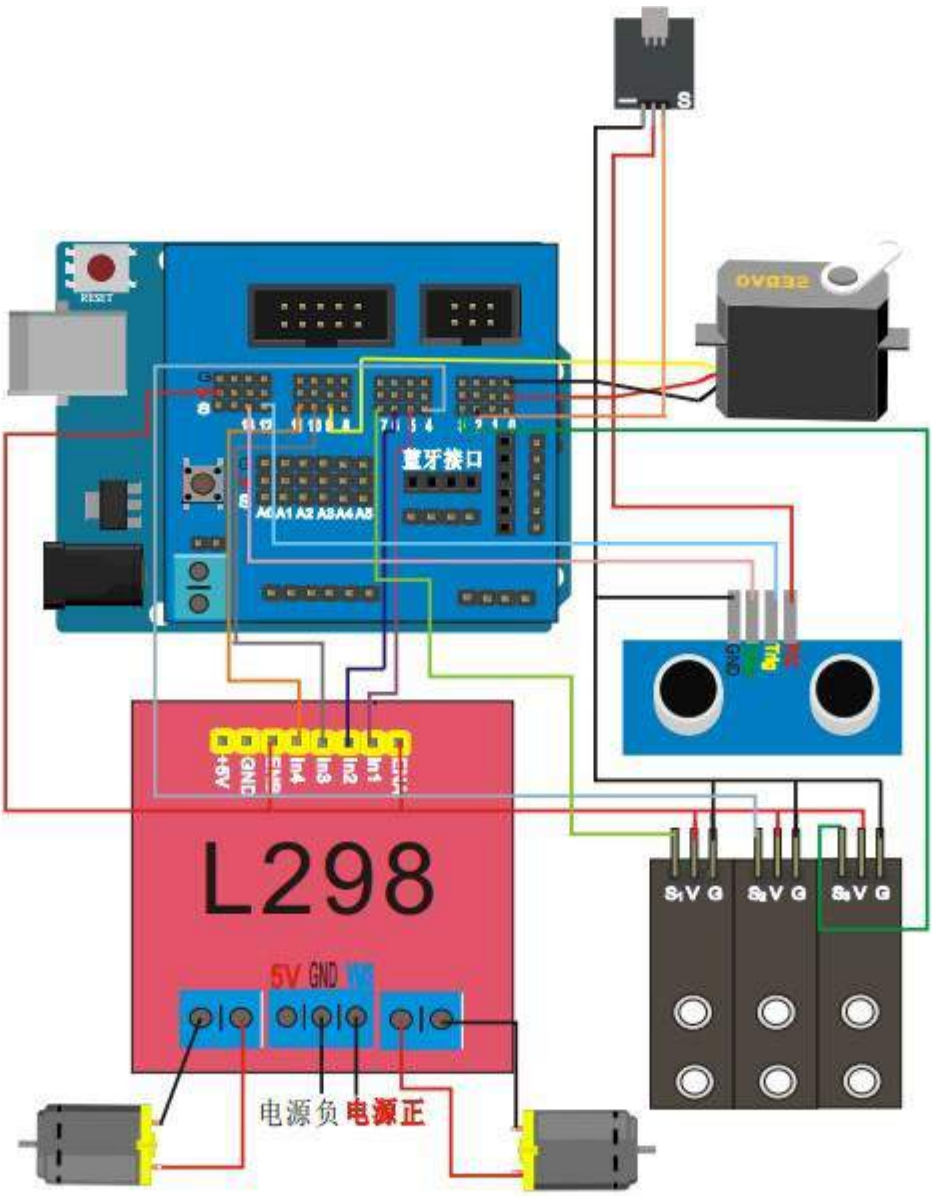
```
#include <LiquidCrystal_I2C.h>  //包含 I2C-LCD1602 液晶的库
#include <Wire.h>              //包含 I2C 的库 unsigned char
Bluetooth_val;     //定义变量 val LiquidCrystal_I2C
lcd(0x27,16,2); //定义一个液晶
#define Lpwm_pin 5  //在這調速 #define
Rpwm_pin 10 //在這調速 // int pinLB=2;
// 定義 2 腳位 左後 int pinLF=4;    // 定
義 4 腳位 左前 int pinRB=7;   // 定義 7
腳位 右後
int pinRF=8;    // 定義 8 腳位 右前
unsigned char Lpwm_val = 150;
unsigned char Rpwm_val = 150; int
Car_state=0;
void M_Control_IO_config(void)
{
  pinMode(pinLB,OUTPUT);    //   腳  位     2
pinMode(pinLF,OUTPUT);    //   腳  位     4
pinMode(pinRB,OUTPUT);    //   腳  位     7
pinMode(pinRF,OUTPUT);    //   腳  位     8
pinMode(Lpwm_pin,OUTPUT); // 腳位 11 (PWM)
  pinMode(Rpwm_pin,OUTPUT); // 腳位 10 (PWM)
}
void Set_Speed(unsigned char Left,unsigned char Right)
{
  analogWrite(Lpwm_pin,Left);
  analogWrite(Rpwm_pin,Right);
}
void advance()    // 前進
   {
    digitalWrite(pinRB,LOW); // 使馬達（右後）動作
    digitalWrite(pinRF,HIGH);
    digitalWrite(pinLB,LOW); // 使馬達（左後）動作
    digitalWrite(pinLF,HIGH);
Car_state = 1;     show_state();
}
void turnR()      //右轉(雙輪)
   {
    digitalWrite(pinRB,LOW); //使馬達（右後）動作
    digitalWrite(pinRF,HIGH);
digitalWrite(pinLB,HIGH);
    digitalWrite(pinLF,HIGH); //使馬達（左前）動作
    Car_state = 4;
show_state();    }
void turnL()      //左轉(雙輪)
```

```
   {
    digitalWrite(pinRB,HIGH);
digitalWrite(pinRF,HIGH);   //使馬達（右前）動作
    digitalWrite(pinLB,LOW);     //使馬達（左後）動作
digitalWrite(pinLF,HIGH);    Car_state = 3;    show_state();
}    void stopp()        //停止
   {
    digitalWrite(pinRB,HIGH);     digitalWrite(pinRF,HIGH);     digitalWrite(pinLB,HIGH);
digitalWrite(pinLF,HIGH);     Car_state = 5;     show_state();    }
void back()         //後退
   {
    digitalWrite(pinRB,HIGH);  //使馬達（右後）動作
digitalWrite(pinRF,LOW);     digitalWrite(pinLB,HIGH);
//使馬達（左後）動作     digitalWrite(pinLF,LOW);
Car_state = 2;     show_state() ;
   }
void show_state(void)
{   lcd.setCursor(0,
1);
switch(Car_state)
  {
    case 1:lcd.print(" Go  ");     break;
    case 2:lcd.print("Back ");     break;
    case 3:lcd.print("Left ");     break;
    case 4:lcd.print("Right");     break;
    case 5:lcd.print("stop ");
    break;
default:
break;    }
}
void LCD1602_init(void)          //液晶初始化用到的函数，包罗在这里
{
 lcd.init();  //调用  LiquidCrystal_I2C.h 里的  LCD 初始化函数
delay(10);   //延时  10 毫秒   lcd.backlight(); //开  LCD1602 的背
光灯   lcd.clear();    //清除屏幕
}
void Bluetooth_Control()
{
  lcd.setCursor(0, 0);  //光标设置在第一行，第一列，也就是左上角
lcd.print("BluetoothControl");   while(Bluetooth_val!= 'E')
  {
  if(Serial.available()) //这一句是用来判断串口是否接收到数据的标志位。
   {
    Bluetooth_val=Serial.read();  //读取（蓝牙）串口数据，并把值赋给  val;
switch(Bluetooth_val)
```

```
    {
      case '8':advance(); Serial.print("\n Car GO ");//UP
break;        case '2': back();   Serial.print("\n Car back ");
//back        break;        case '4':turnL();   Serial.print("\n Car
Left"); //Left        break;
      case '6':turnR();   Serial.print("\n Car Right "); //Right
break;        case '5':stopp();   Serial.print("\n Car stop "); //stop
break;        default:stopp();        break;
    }
   }
  }
  lcd.setCursor(0, 0);   //光标设置在第一行，第一列，
lcd.print("   Wait  Sigal");    stopp();
}
void setup()
{
  LCD1602_init();
  M_Control_IO_config();
  Set_Speed(Lpwm_val,Rpwm_val);
  Serial.begin(9600);   //初始化串口，其实蓝牙已经充当串口的角色。设置波特率为 9600
  lcd.setCursor(0, 0);   //光标设置在第一行，第一列，也就是左上角
lcd.print("   Wait  Sigal");    stopp();
}
void loop()
{
  if(Serial.available()) //这一句是用 8 来判断串口是否接收到数据的标志位。
  {
    Bluetooth_val=Serial.read();
    if(Bluetooth_val=='B') Bluetooth_Control();     }
 delay(20);
}
```

12. Five in One (tracking, infared remote control, Bluetooth control, 1602), Multifunction Program

Wiring diagram:




```
//*******************************
#include <LiquidCrystal_I2C.h>  //包含 I2C-LCD1602 液晶的库
#include <Wire.h>               //包含 I2C 的库
#include <IRremote.h>           //包含遥控器的库
#define RECV_PIN 12             //红外遥控接收器的引脚---12 IRrecv
irrecv(RECV_PIN);       //定义了一个红外遥控器--引脚为 12 decode_results
results;        //红外遥控器的键码缓存
#define IR_Go      0x00ff629d  //前进
#define IR_Back     0x00ffa857  //后退
#define IR_Left     0x00ff22dd  //左转
#define IR_Right    0x00ffc23d  //右转 #define
IR_Stop     0x00ff02fd  //停止
#define IR_ESC      0x00ff52ad  //推出遥控器控制
#define IR_Servo_L  0x00ff6897  //舵机左转
#define IR_Servo_R  0x00ffb04f  //舵机右转
```

```
#define IR_Utr_Ran  0x00ff9867  //超声波测距
#define IR_Speed_UP     0x00ff7a85//车速增加
#define IR_Speed_DOWN   0x00ff30cf//车速减小
#define IR_XunJi_Mode   0x00ff18e7//循迹模式
#define IR_BiZhang_Mode 0x00ff4ab5//避障模式
//////////////////////////////////////////////////
#define SensorLeft    6   //循迹模块左侧传感器引脚
#define SensorMiddle  9   //循迹模块中间侧传感器引脚 #define
SensorRight   11   //循迹模块右侧侧传感器引脚 unsigned char SL;
//循迹左侧传感器状态 unsigned char SM;       //循迹中间传感器状态
unsigned char SR;        //循迹右侧传感器状态 unsigned char
old_SL,old_SM,old_SR;//之前的传感器状态，用于循迹
int inputPin=A0;  // 超声波模块  ECHO to A0 int
outputPin=A1; // 超声波模块  TRIG to A1  unsigned
char Bluetooth_val;       //定义变量  val
LiquidCrystal_I2C lcd(0x27,16,2);  //定义一个液晶
#define Lpwm_pin  5     //车速控制引脚----电机驱动板的  ENA #define
Rpwm_pin  10    //车速控制引脚----电机驱动板的  ENB int pinLB=2;
//车轮转向控制----电机驱动板的  IN1 int pinLF=4;           //车轮转向控
制----电机驱动板的  IN2 int pinRB=7;           //车轮转向控制----电机驱
动板的  IN3 int pinRF=8;           //车轮转向控制----电机驱动板的  IN4
unsigned char Lpwm_val = 150;//初始化左轮速度为  150 unsigned char
Rpwm_val = 150;//初始化右轮速度为  150 int Car_state=0;           //车
子运行状态 int servopin=3;             //定义数字接口  9 连接伺服舵机信
号线
int myangle;                //定义角度变量 int pulsewidth;             //定义脉
宽变量 unsigned char DuoJiao=90;    //舵角初始化为  90 度，也就是
正前方
void Sensor_IO_Config()     //三个循迹模块的  IO 初始化函数，都初始化为输入
{
  pinMode(SensorLeft,INPUT);   pinMode(SensorMiddle,INPUT);
  pinMode(SensorRight,INPUT);
}
void Sensor_Scan(void) //循迹模块的信号读入函数
{
  SL = digitalRead(SensorLeft);
  SM = digitalRead(SensorMiddle);
  SR = digitalRead(SensorRight);
}
void Xunji_Mode(void) //小车循迹模式运行函数
{
   unsigned long IR_ReadData;
   lcd.setCursor(0, 0);  //光标设置在第一行，第一列，也就是左上角
lcd.print("Xun Ji        ");    Show_V(Lpwm_val);
```

```
 while((Bluetooth_val!= 'E')&&(IR_ReadData!=IR_ESC))//当遥控器按下“#”号键，或蓝牙接收到“E”字符，就推出当前模式
 {   if(Serial.available())Bluetooth_val=Serial.read();   if
(irrecv.decode(&results)) IR_ReadData = results.value;
irrecv.resume();
  Sensor_Scan();

if((SL==1&&SM==1&&SR==1)||(SL==1&&SM==1&&SR==0)||(SL==0&&SM==1&&SR==1)
)advance();
 if(SL==1&&SM==0&&SR==0)turnL();   if(SL==0&&SM==0&&SR==1)turnR();
 if(SL==0&&SM==0&&SR==0)
 {
  if((old_SL==1&&old_SM==0&&old_SR==0)||(old_SL==1&&old_SM==1&&old_SR==0))
  {
   while(!(SL==1&&SM==1&&SR==0))
   {
   Sensor_Scan();      turnL();
   }    }
else
if((old_SL==0&&old_SM==0&&old_SR==1)||(old_SL==0&&old_SM==1&&old_SR==1))
  {
   while(!(SL==0&&SM==1&&SR==1))
   {
     Sensor_Scan();        turnR();
   }
  }
 }
 if(old_SL!=SL||old_SM!=SM||old_SR!=SR)
 {    old_SL = SL;
old_SM = SM;
  old_SR = SR;                        }
 }//代码运行到这，说明将要推出此模式，所以要清屏，停车。
lcd.clear();
  lcd.setCursor(0, 0);  //光标设置在第一行，第一列，
lcd.print("  Wait  Signal  ");   stopp();
}
void servopulse(int servopin,int myangle)//定义一个脉冲函数
{
pulsewidth=(myangle*11)+500;//将角度转化为 500-2480 的脉宽值
digitalWrite(servopin,HIGH);//将舵机接口电平至高 delayMicroseconds(pulsewidth);//延时脉宽值的微秒数
digitalWrite(servopin,LOW);//将舵机接口电平至低
delay(20-pulsewidth/1000);
} void Set_servopulse(int
set_val)
```

```
{
 for(int i=0;i<=10;i++) //给予舵机足够的时间让它转到指定角度
   servopulse(servopin,set_val);//引用脉冲函数
}
void M_Control_IO_config(void)
{
  pinMode(pinLB,OUTPUT);      //      腳     位          2
pinMode(pinLF,OUTPUT);      //      腳     位          4
pinMode(pinRB,OUTPUT);      //      腳     位          7
pinMode(pinRF,OUTPUT);      //      腳     位          8
pinMode(Lpwm_pin,OUTPUT); // 腳位 11 (PWM)
  pinMode(Rpwm_pin,OUTPUT); // 腳位 10 (PWM)
}
void Set_Speed(unsigned char Left,unsigned char Right)//车速设定函数，
{
  analogWrite(Lpwm_pin,Left);
  analogWrite(Rpwm_pin,Right);
}
void advance()        // 前进
    {
digitalWrite(pinRB,LOW);
digitalWrite(pinRF,HIGH);
digitalWrite(pinLB,LOW);
digitalWrite(pinLF,HIGH);
Car_state = 1;      show_state();
}
void turnR()           //右转
    {
     digitalWrite(pinRB,LOW);  //使馬達（右後）動作
     digitalWrite(pinRF,HIGH);
digitalWrite(pinLB,HIGH);
     digitalWrite(pinLF,LOW);  //使馬達（左前）動作
     Car_state = 4;
show_state();     }
void turnL()          //左转
    {
     digitalWrite(pinRB,HIGH);
digitalWrite(pinRF,LOW);   //使馬達（右前）動作
     digitalWrite(pinLB,LOW);   //使馬達（左後）動作
     digitalWrite(pinLF,HIGH);
Car_state = 3;      show_state();
}
void stopp()          //停止
    {
digitalWrite(pinRB,HIGH);
```

```
digitalWrite(pinRF,HIGH);
digitalWrite(pinLB,HIGH);
digitalWrite(pinLF,HIGH);
Car_state = 5;      show_state();
}
void back()          //后退
   {
     digitalWrite(pinRB,HIGH);  //使馬達（右後）動作
digitalWrite(pinRF,LOW);      digitalWrite(pinLB,HIGH);
//使馬達（左後）動作      digitalWrite(pinLF,LOW);
Car_state = 2;      show_state() ;        } void
show_state(void)
{   lcd.setCursor(0,
1);
switch(Car_state)    {
case 1:lcd.print(" Go
");Serial.print(" \r\n
GO");      break;
     case 2:lcd.print("Back ");Serial.print(" \r\n Back");      break;
     case 3:lcd.print("Left ");Serial.print(" \r\n Left");      break;
     case 4:lcd.print("Right");Serial.print(" \r\n Right");
break;      case 5:lcd.print("Stop ");Serial.print(" \r\n
Stop");       break;      default:      break;    }
}
void LCD1602_init(void)           //液晶初始化用到的函数，包罗在这里
{
  lcd.init();  //调用  LiquidCrystal_I2C.h 里的  LCD 初始化函数
delay(10);   //延时  10 毫秒   lcd.backlight(); //开  LCD1602 的背
光灯   lcd.clear();     //清除屏幕
}
void Show_V(unsigned char V)
{                lcd.setCursor(11,   0);
lcd.print("V=                      ");
lcd.setCursor(13,                  0);
lcd.print(V,DEC);
Serial.print("\n Speed = ");
     Serial.print(V,DEC);
}
void Show_DuoJiao(unsigned char Jiao)
{                     lcd.setCursor(6,1);
lcd.print("C=                          ");
lcd.setCursor(8,                      1);
lcd.print(Jiao,DEC);
Serial.print("\n JiaoDu = ");
     Serial.print(Jiao,DEC);
}
```

```
void Bluetooth_Control()//蓝牙控制模式，
{
  lcd.setCursor(0, 0); //光标设置在第一行，第一列，也就是左上角    lcd.print("Blue_ctr
");
  Show_V(Lpwm_val);
 while(Bluetooth_val!= 'E')
 {
  if(Serial.available()) //这一句是用来判断串口是否接收到数据的标志位。
   {
    Bluetooth_val=Serial.read(); //读取（蓝牙）串口数据，并把值赋给 val;
switch(Bluetooth_val)
    {
      case '7': if(DuoJiao<=180){       //舵机左转
                DuoJiao+=10;
                Set_servopulse(DuoJiao);
                Show_DuoJiao(DuoJiao);}                  break;
case '9': if(DuoJiao-10>=0){        //舵机右转
                DuoJiao-=10;
                Set_servopulse(DuoJiao);
                Show_DuoJiao(DuoJiao);}
break;        case '8':advance(); //向前        break;
      case '2': back();    //退后        break;
      case '4':turnL();    //左转        break;
      case '6':turnR();    //右转        break;        case '5':stopp();    //停        break;
case '3':if(Rpwm_val+10<=250&&Rpwm_val+10<=250){ //车速减小
                Lpwm_val+=10; Rpwm_val+=10;
                Set_Speed(Lpwm_val,Rpwm_val);
                Show_V(Lpwm_val);
                }       break;       case '1':if(Rpwm_val-10>=0&&Rpwm_val-
10>=0){    //车速增大
                Lpwm_val-=10; Rpwm_val-=10;
                Set_Speed(Lpwm_val,Rpwm_val);
                Show_V(Lpwm_val);
                }       break;
      case '0': Ultrasonic_Ranging(1);   //超神波测距
break;        default:stopp();        break;
    }     }
 }//程序运行到这里说明将准备推出此模式，所以要清屏，停车。
lcd.clear();
  lcd.setCursor(0, 0); //光标设置在第一行，第一列，
lcd.print("  Wait  Signal  ");   stopp();
}
void IR_Control(void)   //遥控器控制模式，当按下“#”号键的时候，退出此模式
```

```
{
   unsigned long Key;
   lcd.setCursor(0,0);  //光标设置在第一行，第一列，也就是左上角
   lcd.print("IR_Ctr          ");    Show_V(Lpwm_val);
   Show_DuoJiao(DuoJiao);//显示当前舵脚
  while(Key!=IR_ESC)
  {
   if(irrecv.decode(&results)) //这一句是用来判断串口是否接收到数据的标志位。
    {
     Key = results.value;      //把键值读出来并传给  Key
switch(Key)      {
       case IR_Servo_L: if(DuoJiao<=180){  //舵机左转
              DuoJiao+=10;
              Set_servopulse(DuoJiao);
              Show_DuoJiao(DuoJiao);}                    break;
       case IR_Servo_R: if(DuoJiao-10>=0){  //舵机右转
              DuoJiao-=10;
              Set_servopulse(DuoJiao);                  Show_DuoJiao(DuoJiao);}
       break;
            case IR_Speed_UP:if(Rpwm_val+10<=250&&Rpwm_val+10<=250){  //车速增加
              Lpwm_val+=10; Rpwm_val+=10;
              Set_Speed(Lpwm_val,Rpwm_val);
              Show_V(Lpwm_val);
              }        break;
       case IR_Speed_DOWN:if(Rpwm_val-10>=0&&Rpwm_val-10>=0){  //车速减小
              Lpwm_val-=10; Rpwm_val-=10;
              Set_Speed(Lpwm_val,Rpwm_val);
              Show_V(Lpwm_val);                  }
       break;
       case IR_Utr_Ran: Ultrasonic_Ranging(1);  //超声波测距        break;
       case IR_XunJi_Mode:Xunji_Mode();        break;
       case IR_BiZhang_Mode:Self_Control();        break;
       case IR_Go:advance();    //前进        break;
       case IR_Back: back();    //后退
break;        case IR_Left:turnL();    //左转
break;
       case    IR_Right:turnR();       //  右  转
break;       case IR_Stop:stopp();    //停止
break;        default:         break;
     }
    irrecv.resume(); //清除当前红外键值，为下一次读取键值做好准备
   }    }
lcd.clear();
```

```
 lcd.setCursor(0, 0); //光标设置在第一行，第一列，
lcd.print("  Wait  Signal  ");  stopp();
}
void Self_Control(void)//自走，及超声波避障
{   int H;
  lcd.setCursor(0, 0); //光标设置在第一行，第一列，也就是左上角
lcd.print("Self_Ctr       ");   Show_V(Lpwm_val);
  while(Bluetooth_val!= 'E')
  {
    if(Serial.available()) //这一句是用来判断串口是否接收到数据的标志位。
Bluetooth_val=Serial.read(); //读取（蓝牙）串口数据，并把值赋给 val;
    H = Ultrasonic_Ranging(1);     if(H<25)
{        stopp();
while(Ultrasonic_Ranging(1)<50)
       {
         if(Ultrasonic_Ranging(1)<15)
         {            back();
delay(500);          }
else           {
turnL();
delay(1000);          }
       }
    } advance();
  }   lcd.clear();
  lcd.setCursor(0, 0); //光标设置在第一行，第一列，    lcd.print("
Wait  Signal  ");
  stopp();
}
int Ultrasonic_Ranging(unsigned char Mode)//超声波测距函数，MODE=1 时，显示，其他不显示。
{
 int old_distance;   digitalWrite(outputPin, LOW);
delayMicroseconds(2);    digitalWrite(outputPin, HIGH); //
delayMicroseconds(10);    digitalWrite(outputPin, LOW);      int
distance = pulseIn(inputPin, HIGH);  // 读取高电平持续时间。
distance= distance/58;   // Transform pulse time to distance
if(Mode==1){         lcd.setCursor(11, 1);         lcd.print("H=    ");
lcd.setCursor(13, 1);         lcd.print(distance,DEC);
Serial.print("\n H = ");         Serial.print(distance,DEC);         return
distance;
 }
  else  return distance;
}  void
setup()
{
  pinMode(servopin,OUTPUT);  //设定舵机接口为输出接口
```

```
  LCD1602_init();           //1602 初始化
  M_Control_IO_config();    //电机控制模块 IO 初始化
  Set_Speed(Lpwm_val,Rpwm_val); //初始速度设定
Set_servopulse(DuoJiao);      //舵角初始设定   Sensor_IO_Config();
//循迹模块 IO 初始化   irrecv.enableIRIn();         //开始接收红外遥控器
信号   pinMode(inputPin, INPUT);     //超声波模块 IO
pinMode(outputPin, OUTPUT);   //超声波模块 IO
  Serial.begin(9600);          //初始化串口，其实蓝牙已经充当串口的角色。设置波特率为
9600
  lcd.setCursor(0, 0);          //光标设置 0.0   lcd.print("
Wait  Signal  "); //lcd 打印字符串   stopp();                    //
停车
}  void
loop()
{
  if(Serial.available()) //这一句是用 8 来判断串口是否接收到数据的标志位。
  {
    Bluetooth_val=Serial.read();
    if(Bluetooth_val=='W') Bluetooth_Control();//当蓝牙接收到大写"W"后，进入蓝牙控制
模式
    if(Bluetooth_val=='S') Self_Control();    //当蓝牙接收到大写"S"后，进入超声波避障模
式
    if(Bluetooth_val=='X') Xunji_Mode();      //当蓝牙接收到大写"X"后，进入循迹模式
   }
  if (irrecv.decode(&results)) {//当收到一个按键      if(results.value == IR_Stop )IR_Control();  //
当遥控器按下"OK"键的时候，进入遥控器控制模式
    irrecv.resume(); //清除当前红外键值，为下一次读取键值做好准备
   }    delay(10);
}
```